

UNIVERSITÄT PASSAU
Fakultät für Mathematik und Informatik

Dissertation

**Interaktives orthogonales Zeichnen von
planaren Graphen**

Walter Bachl

Passau, Juni 2003

Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften an der
Fakultät für Mathematik und Informatik der Universität Passau.

1. Gutachter: Prof. Dr. Franz J. Brandenburg, Universität Passau
2. Gutachter: Prof. Dr. Michael Kaufmann, Universität Tübingen

Inhaltsverzeichnis

1	Einleitung	7
2	Allgemeine Definitionen	10
2.1	Graphen und ihre Zeichnungen	10
2.2	Orthogonale Graphen im interaktiven Kontext	13
3	Orthogonale und interaktive Zeichenmodelle und -verfahren	16
3.1	Statische orthogonale Zeichenmodelle	16
3.1.1	Punktförmige Knoten	16
3.1.2	Zeichenmodell für unbeschränkte Knotengrößen	18
3.1.3	TSS-Modell	20
3.1.4	Kandinsky-Modell	21
3.2	Interaktive Zeichenverfahren	23
4	Theoretische Grundlagen	29
4.1	Restriktionen für Zeichnungen	30
4.2	Änderungen an der Graphstruktur	32
4.3	Qualitätskriterien	34
4.4	Theoretisch erreichbare Ergebnisse	35
5	Einfügeoperationen	45
5.1	Einfügeoperationen	45
5.1.1	Erzeugen eines initialen Graphen	46
5.1.2	Einfügen eines Knotens	46
5.1.3	Einfügen einer Kante	47
5.2	Lineare Laufzeit	51
5.2.1	Verwendete Datenstrukturen	51
5.2.2	Einfügen eines Knotens	53
5.2.3	Einfügen einer Kante	54
5.3	Erweiterungen für logarithmische Laufzeit	55
5.3.1	Sektorbäume	58
5.3.2	Attribute von Knoten, Kanten und Flächen	59
5.3.3	Einfügen eines Knotens	61
5.3.4	Einfügen einer Kante	63

6	Erzeugung einer orthogonalen Repräsentation	66
7	Minimierung der Anzahl an Kantenknicken	78
7.1	Rahmenbedingungen für Knickeinsparungen	78
7.2	Worst-Case Konstruktionen zur Knickeinsparung	80
7.3	Knickeinsparungen bei einzelnen Kanten	85
7.4	Statische Optimierung	97
7.4.1	Waage- und senkrechte Kanten ohne Knicke	98
7.4.2	Die Einsparung eines Knickes bei Kanten mit zwei Knicken	107
7.4.3	Zusammenfassung	138
7.5	Interaktive Knickoptimierung	140
8	Zeichnen des Graphen	145
8.1	Zeichnen mit minimaler Fläche ist NP-vollständig	145
8.2	Algorithmus zum Berechnen einer Zeichnung	161
9	Erweiterungen des Basisalgorithmus	167
9.1	Benutzerdefinierte Knotengrößen	167
9.2	Kantenbeschriftungen	168
9.2.1	Kanten mit einem Knick	170
9.2.2	Kanten mit zwei Knicken	171
9.2.3	Kanten ohne Knick	173
9.3	Einfach zusammenhängende Graphen	176
9.3.1	Zeichnen eines einfach zusammenhängenden Graphen	177
9.3.2	Verschmelzen von zwei zweifach Zusammenhangskomponenten . .	180
9.4	Nicht planares Einfügen von Kanten	182
10	Zusammenfassung und Ausblick	186

1 Einleitung

In der Geschäftswelt und der Forschung werden viele Vorgänge und Sachverhalte graphisch modelliert. Beispiele hierfür sind Entity-Relationship-Diagramme, die Darstellung einer prozentualen Verteilung in Tortendiagrammen, „visuelle“ Programmierung mit Hilfe von UML-Diagrammen, Repräsentation von Firmenhierarchien, Verzeichnisstrukturen, Funktionsgraphen zur Illustration eines Wachstumsverhaltens, Biochemische Pfadwege oder Soziogramme zur Darstellung der sozialen Struktur in einer Gruppe von Menschen. Die visuelle Darstellung hat stets einen Grund: Das Studium von „nackten Zahlenkolonnen“ ist für den Menschen meist sehr mühselig. Deshalb wird gemäß dem Sprichwort „Ein Bild sagt mehr als tausend Worte“ eine Grafik verwendet, um komplexe Zusammenhänge verständlicher darzustellen.

Viele graphische Modellierungen haben eine Gemeinsamkeit: Sie stellen Objekte und Zusammenhänge dar. Unter den obigen Beispielen sind dies Entity-Relationship-Diagramme, UML-Diagramme, Firmenhierarchien, Verzeichnisstrukturen, Zeichnungen von Computerchips, Biochemische Pfadwege und Soziogramme. Objekte werden durch verschiedenste Symbole visualisiert, Beziehungen zwischen den Objekten durch Pfeile oder Verbindungslinien. Die mathematische, theoretisch fundierte Entsprechung dazu ist die Darstellung mittels eines *Graphen*. Ein Graph besteht aus einer Menge von *Knoten* — den Objekten — und einer meistens zweistelligen Relation zwischen den Objekten, den *Kanten*. Die Kanten stellen die Beziehungen zwischen den Objekten dar. Mehrstellige Relationen lassen sich mit Hypergraphen oder bipartiten Graphen beschreiben.

Selbstverständlich genügt es nicht, eine beliebige graphische Repräsentation der realen Welt zu finden. Die Darstellung muss die Realität der Aufgabenstellung entsprechend abbilden und, ebenso wichtig, sie muss so geartet sein, dass sie für den Menschen leicht verständlich und schön ist. Wenn es gelingt, Algorithmen zum Zeichnen von Graphen zu entwickeln, dann stehen diese in allen Anwendungen, in denen die graphische Repräsentation auf Objekte und deren Beziehungen aufbaut, zur Verfügung. Aufgrund dieser Bedeutung in den verschiedensten Disziplinen ist das automatische Zeichnen von Graphen ein lebhaftes und aktives Forschungsgebiet.

Die generierten Zeichnungen sollen dabei Kriterien wie „Schönheit“ oder „leichte Verständlichkeit“ erfüllen. „Schön“ und „leicht verständlich“ sind dabei zwei subjektive Begriffe, denn jeder Mensch hat eine andere Vorstellung davon, was schön oder leicht verständlich ist. Aus diesem Grund lassen sich diese Begriffe nur schwer formalisieren. Trotzdem hat sich beim Zeichnen von Graphen eine Menge von allgemein anerkannten Ästhetikkriterien herausgebildet (vgl. [PCJ95]), von denen angenommen wird, dass

sie die Begriffe „schön“ und „leicht verständlich“ beschreiben. Teilweise sind diese Annahmen durch Studien abgesichert. Zu den Ästhetikkriterien zählen beispielsweise die Unterschiede in den Kantenlängen, die Anzahl der Schnitte zwischen Kanten oder die gleichmäßige Verteilung der Knoten auf einer Fläche.

Die Modellierung der realen Welt kann auf zwei verschiedene Arten entstehen: Es existiert ein Computerprogramm und dieses generiert z. B. aus einer Datenbank das entsprechende Entity-Relationship-Diagramm. Dafür gibt es sowohl für spezielle Graphklassen als auch für allgemeine Graphen mehrere Zeichenverfahren. Andererseits kann ein Graph aber auch dadurch entstehen, dass ein Benutzer den Graphen interaktiv am Computerbildschirm erstellt. Dies geschieht zum Beispiel normalerweise bei der Erstellung von UML-Diagrammen. Wir wollen ein solches Szenario näher betrachten.

Wenn ein Graph interaktiv erstellt wird, dann kennt der Benutzer in der Regel am Anfang den Graphen nicht, der am Ende entstehen soll. Dies führt zu einem gravierenden Problem: Da dem Benutzer das endgültige Resultat nicht bekannt ist, kann er zukünftige Objekte, die später zusätzlich dargestellt werden müssen, nicht von Anfang an mit einplanen. Als Folge erhält er früher oder später eine Situation, in der das nächste Objekt nicht mehr an einer sinnvollen Stelle am Bildschirm platziert werden kann.

Man kann in diesem Fall entweder mit einem Zeichenalgorithmus eine neue Zeichnung erstellen oder man strukturiert die Zeichnung von Hand neu. Im ersten Fall kann das Computerprogramm die Objekte des Graphen beliebig neu am Bildschirm anordnen, wodurch man gezwungen ist, sich in der Zeichnung neu zu orientieren. Im zweiten Fall ist die manuelle Anordnung mit zunehmender Größe des Graphen immer umfangreicher. In beiden Fällen ist man also gezwungen, einen zeitaufwendigen Vorgang in Kauf zu nehmen. Wenn sich dieser Vorgang mehrfach wiederholt, dann ist das zeitraubend und nervend. Der Effekt wird auch dadurch verstärkt, dass die Diagramme im Normalfall in einem Top-Down-Design erstellt werden. Die lokale Verfeinerung an einer Stelle führt zu einer ungleichmäßigen Verteilung der Knoten. An Stellen, an denen sich die Knoten häufen, ist bald kein Platz für weitere Knoten.

Deshalb ist es sinnvoll, auch für diesen Zweck automatische Zeichenverfahren anzubieten, die die Zeichnung immer so weit anpassen, dass das weitere Einfügen von Objekten an jeder beliebigen Stelle möglich ist. Die vorhandenen Algorithmen, die von statischen Graphen ausgehen, sind dabei ungeeignet. Sie nehmen keine Rücksicht auf eine schon vorhandene Zeichnung und deshalb kann eine kleine Änderung am Graphen zu einer großen Änderung an der Zeichnung führen. Dies ist unerwünscht, da man sich während der Erzeugung des Graphen die Lage der einzelnen Objekte merkt, man schafft sich ein geistiges Abbild der Zeichnung, im Fachjargon „mental map“ (siehe z.B. [ELMS91]). Durch die neue Zeichnung ist der Benutzer gezwungen, sich neu zu orientieren, was zeitaufwendig und damit nicht gewollt ist.

Es ist also sinnvoll, sich spezielle Verfahren für das interaktive Zeichnen von Graphen zu überlegen, die versuchen, die vorhandene Zeichnung bestmöglich zu erhalten. Gleichzeitig muss es aber in einem gewissen Rahmen erlaubt sein, dass sich die Zeichnung ändert, da sonst die Qualität der generierten Zeichnungen zu sehr leidet. Hier ist also ein Kompromiss zwischen einer guten Zeichnung und minimaler Änderung zu finden.

Im Rahmen der vorliegenden Arbeit beschäftigen wir uns mit einem speziellen Zeichenstil: Interaktives orthogonales Zeichnen. Orthogonal heißt, dass eine Kante aus einer Sequenz von waagerechten und senkrechten Linienstücken besteht. Diese Art von Zeichnung wirkt für viele Menschen besonders „aufgeräumt“ und damit übersichtlich. Auch beim VLSI-Design werden orthogonale Zeichnungen erzeugt. Bei der Berechnung der Zeichnungen achten wir insbesondere darauf, dass die Änderung in aufeinander folgenden Zeichnungen so strukturiert ist, dass diese von Benutzern nicht als störend empfunden wird. Dennoch gelingt es, die Anzahl der Knicke in den Kanten relativ klein zu halten. Die Anzahl der Kantenknicke gilt als wichtiges Qualitätskriterium für Algorithmen zum orthogonalen Zeichnen von Graphen.

Das in dieser Arbeit entwickelte Verfahren gliedert sich in einen Basisalgorithmus und eine Menge von weitgehend unabhängigen zusätzlichen Komponenten.

Der Basisalgorithmus ist für zweifach zusammenhängende, planare Graphen konzipiert. Da sich die Zeichnung eines Graphen, wie vorhin gefordert, in einem interaktiven Kontext nicht beliebig ändern darf, basiert der Algorithmus auf einem System von Lagebeziehungen zwischen den Knoten des Graphen und seiner Einbettung. Sie bilden die Invarianten des Algorithmus. Wir erlauben deshalb nur Ergänzungen, aber keine Änderungen daran. Damit wird garantiert, dass aufeinander folgende Zeichnungen des Graphen einander ähnlich sind. Die Funktionen zum sukzessiven Aufbau der zweifach zusammenhängenden, planaren Graphen sind *InsertNode* und *InsertEdge*. Das Einfügen einer Kante ist dabei nur dann möglich, wenn die Endknoten in einer gemeinsamen Fläche liegen. Beim Erstellen einer Zeichnung aus der Einbettung und den Lagebeziehungen wird in einem Zwischenschritt zunächst noch eine orthogonale Repräsentation erzeugt.

Man erhält damit für zweifach zusammenhängende planare Graphen einen interaktiven Zeichenalgorithmus, mit dem man alle Graphen dieser Graphklasse darstellen kann. Alle anderen Komponenten sind Ergänzungen, die beliebig und voneinander unabhängig hinzugefügt werden können. Sie dienen zur Erweiterung der betrachteten Graphklasse (nicht planare Graphen, einfacher Zusammenhang), erlauben in der Praxis notwendige Ergänzungen (beliebige Knotengrößen, Kantenbeschriftungen) und verbessern die Qualität der Zeichnung (wenige Kantenknicke).

Im Folgenden wollen wir nach einer formalen Definition aller wichtigen Begriffe zunächst die bereits in der Literatur vorhandenen Arbeiten analysieren. Ausgehend von diesen Arbeiten beschäftigen wir uns dann mit orthogonalen statischen Zeichenverfahren, um die theoretisch erreichbaren Grenzen für ein interaktives Zeichenverfahren abzuleiten. Danach wird der Basisalgorithmus für zweifach zusammenhängende, planare Graphen entwickelt, der dann sukzessive um weitere Fragestellungen, die sich bei realen Anwendungen ergeben, erweitert wird. Abschließend wird die Arbeit zusammengefasst und aus unserer Sicht interessante zusätzliche Fragen werden noch angesprochen.

2 Allgemeine Definitionen

In diesem Kapitel werden die für diese Arbeit nötigen Definitionen angegeben. Dazu beschäftigen wir uns zunächst mit Graphen und orthogonalen Zeichnungen. Danach werden die Begriffe und Notationen erklärt, die zur Beschreibung des schrittweisen Aufbaus des Graphen verwendet werden. Weiterführende Definitionen zur Beschreibung von Graphen und Graphalgorithmen findet man z.B. in [BETT98] oder [Har74].

2.1 Graphen und ihre Zeichnungen

Ein *Graph* besteht aus einer Menge von Knoten und einer Menge von Kanten. Dabei verbindet eine Kante immer zwei Knoten. Im Folgenden bezeichne $G = (V, E)$ den Graphen G mit der Knotenmenge V und der Kantenmenge E . n bezeichne die Anzahl der Knoten und m die Anzahl der Kanten in $G = (V, E)$. Wenn V und E aus dem Kontext ersichtlich sind, dann verwenden wir auch G anstelle von $G = (V, E)$.

Ein Graph heißt *gerichtet*, falls jede Kante einen eindeutigen *Quellknoten* und *Zielknoten* besitzt, d.h. $E = \{(v_i, v_j) \mid v_i, v_j \in V\} \subseteq V \times V$. Andernfalls heißt der Graph *ungerichtet*. Hier verbindet eine Kante zwei Knoten, ohne dabei einen Quell- oder Zielknoten zu definieren, d.h. $E = \{\{v_i, v_j\} \mid v_i, v_j \in V\}$. Falls nicht anders erwähnt, gehen wir in der Arbeit stets von ungerichteten Graphen aus. Alle Definitionen und Ergebnisse dieser Arbeit können aber auf gerichtete Graphen übertragen werden, wenn man je einen Endknoten einer Kante zu einem Quell- bzw. Zielknoten macht.

Eine Kante heißt *inzident* zu einem Knoten v , wenn v ein Endknoten der Kante ist. Zwei Knoten, die durch eine Kante verbunden sind, nennt man *adjazent* oder auch *benachbart*. Unter dem *Grad* eines Knotens verstehen wir die Anzahl seiner inzidenten Kanten. Ein Folge von Knoten $[v_1, \dots, v_k]$ heißt *Pfad* zwischen den Knoten v_1 und v_k , wenn für alle $i \in \{1, \dots, k-1\}$ gilt: $\{v_i, v_{i+1}\} \in E$.

Ein Graph heißt *bipartit*, falls sich seine Knotenmenge in zwei Teilmengen X und Y partitionieren lässt, so dass es keine Kante gibt, die zwei Knoten in X oder zwei Knoten in Y verbindet. Wir schreiben kurz $G = (X \cup Y, E)$ für einen bipartiten Graphen.

Es existieren viele Möglichkeiten, Graphen in unterschiedliche Klassen einzuteilen. Eine davon ist die Klassifikation anhand des Zusammenhangs. Ein Graph heißt *zusammenhängend*, wenn je zwei Knoten durch einen Pfad verbunden sind. Ein Graph G heißt *k-fach zusammenhängend*, wenn der Graph nach dem Entfernen einer beliebigen $k-1$ -elementigen Teilmenge der Knoten immer noch zusammenhängend ist.

Bei einer *Zeichnung* eines Graphen wird seinen Knoten eine Position und seinen Kanten ein ununterbrochener Linienzug im \mathbb{R}^2 zugewiesen, der die beiden Endknoten verbindet.¹ Oft wird auch der Begriff *Layout* oder *Diagramm* synonym für Zeichnung benutzt. Im Rahmen dieser Arbeit besitzt jeder Knoten stets eine positive Höhe und Breite. Wir lösen uns also an dieser Stelle von Zeichenmodellen, in denen ein Knoten keine Ausdehnung² in der Fläche hat. Wir bezeichnen die x-Koordinate des Mittelpunktes eines Knotens v mit $v.x$ und seine y-Koordinate mit $v.y$. Die Höhe und Breite eines Knotens erhalten wir durch $v.h$ bzw. $v.w$. Außerdem setzt sich eine Kante aus einer Sequenz miteinander verbundener Geradenstücke zusammen. Der Verlauf einer Kante $e = \{v_1, v_2\}$ wird durch eine Sequenz $e.line() = (p_1, \dots, p_n)$ von Punkten angegeben. Je zwei aufeinander folgende Punkte werden durch eine Gerade verbunden. Unter $L(G)$ verstehen wir die Zeichnung des gesamten Graphen, also die Belegung der Attribute x, y, w und h für einen Knoten und für eine Kante mit *line* die Festlegung des Kantenverlaufes. Die Zeichnung ist also durch die Koordinaten der Knoten, ihre Höhe und Breite und die Kantenverläufe festgelegt.

Die *Zeichnung* eines Graphen ist *planar*, wenn in der Zeichnung keine Knoten- und Kantenschnitte auftreten. Ein Graph G heißt *planar*, wenn eine planare Zeichnung von G existiert. Unter einer *Fläche* einer planaren Zeichnung verstehen wir einen maximalen zusammenhängenden Teil des \mathbb{R}^2 , der frei von Knoten und Kanten ist. Die Knoten und Kanten einer Fläche sind diejenigen Knoten und Kanten, die die Fläche begrenzen. Es existiert genau eine unbegrenzte Fläche, die als *Außenfläche* bezeichnet wird.

Planare Graphen können auch strukturell durch die Angabe von verbotenen Subgraphen charakterisiert werden. Dabei hilft der Satz von Kuratowski (siehe z. B. [Har74]):

Satz von Kuratowski

Ein Graph ist genau dann planar, wenn er keinen Subgraphen enthält, der homeomorph zum K_5 oder zum $K_{3,3}$ ist. □

Dabei ist der K_5 der vollständige Graph³ mit fünf Knoten und der $K_{3,3}$ ein vollständiger bipartiter Graph, wobei beide Teilmengen der Knotenmenge je drei Knoten enthalten. Diese beiden Graphen werden auch als *Kuratowski-Graphen* bezeichnet.

Wenn man für einen planaren Graphen die Flächen und die Abfolge der Knoten und Kanten, die jede Fläche begrenzen, festlegt, dann bezeichnet man das als *planare Einbettung* des Graphen. Eine planare Einbettung ist eindeutig durch die zyklische Ordnung der Kanten um ihre Endknoten und die Wahl der Außenfläche festgelegt. Im Laufe der Arbeit verwenden wir oft diese Ordnung der Kanten um einen Knoten. Dabei können die Kanten im oder gegen den Uhrzeigersinn durchlaufen werden. Wenn die Reihenfolge nicht explizit erwähnt wird, dann ist die Ordnung im Uhrzeigersinn gemeint. Außerdem müssen oft die Knoten und Kanten um eine Fläche betrachtet werden. Auch hier kann das im oder gegen den Uhrzeigersinn geschehen und wir verwenden ebenfalls die Ordnung im Uhrzeigersinn, falls die Reihenfolge nicht explizit angegeben ist.

¹ Manchmal wird auch der \mathbb{R}^3 für Zeichnungen verwendet. Mit solchen drei-dimensionalen Zeichnungen beschäftigen wir uns hier aber nicht.

² In diesem Fall spricht man von punktförmigen Knoten.

³ Ein Graph ist vollständig, wenn jeder Knoten mit allen anderen durch eine Kante verbunden ist.

Es gibt mehrere planare Einbettungen für einen Graphen, und für jede planare Einbettung existieren beliebig viele planare Zeichnungen. In Abbildung 2.1 wird dieser Sachverhalt illustriert. Man kann die Wahl der planaren Einbettung als Äquivalenzklassenbildung auf der Menge aller planaren Zeichnungen eines Graphen betrachten. Wenn bei einem dreifach zusammenhängenden Graphen die Außenfläche festgelegt ist, dann ist die Einbettung eindeutig gegeben.

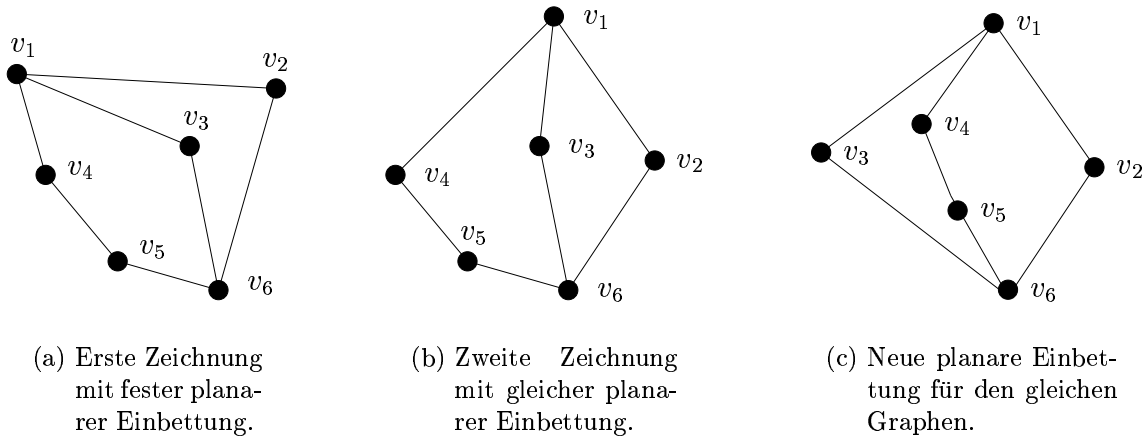


Abbildung 2.1: Ein Graph hat mehrere verschiedene planare Einbettungen und Zeichnungen.

Im Rahmen dieser Arbeit beschäftigen wir uns mit *Gitterzeichnungen*. Eine Gitterzeichnung ist eine Zeichnung, bei der Knotenmittelpunkte auf den Schnittpunkten eines Gitters liegen müssen. Kanten müssen entlang der Gitterlinien verlaufen.

Die Zeichnung einer Kante heißt *orthogonal*, falls alle Kantensegmente horizontal oder vertikal verlaufen.

Eine *orthogonale Repräsentation* eines Graphen legt die Verläufe aller Kanten fest (das heißt, die Abfolge von horizontalen und vertikalen Kantensegmenten), Knoten haben jedoch keine Koordinaten und Kantensegmente keine Längen. Eine orthogonale Repräsentation beschreibt also die Umrisse in einer orthogonalen Zeichnung. Formal kann eine orthogonale Repräsentation z.B. durch eine geordnete Liste von Tupeln (w_e, l_e) für alle Flächen f beschrieben werden, wobei e eine Kante ist, w_e den Winkel zur Vorgängerkante in f beschreibt und l_e eine Sequenz von Knicken der Kante e ist, gelesen im Uhrzeigersinn um f . Man spricht auch vom „topology-shape-metric“ Ansatz, wenn einem Graphen zunächst eine Einbettung, dann Kantenverläufe und schließlich Längen und Positionen zugewiesen werden (siehe [BETT98]).

In [BBD⁺99] wird eine Einschränkung der orthogonalen Repräsentationen von planaren eingebetteten Graphen definiert. Mit *turn regularity* werden orthogonale Repräsentationen charakterisiert, bei denen jedes Knotenpaar eine *orthogonale Relation* hat. Ein Knotenpaar $\{u, v\}$ hat eine orthogonale Relation, wenn durch die orthogonale Repräsentation die relative Lage der beiden Knoten entweder in der Vertikale oder in der

Horizontale festgelegt ist. Das heißt, in allen planaren Zeichnungen, die dieser orthogonalen Repräsentation entsprechen, gilt immer entweder $u.x < v.x$, $u.y < v.y$ oder beides.

Die Bedeutung von „turn-regularity“ ergibt sich aus der Tatsache, dass eine Zeichnung mit minimaler Fläche in Linearzeit berechnet werden kann ([BBD⁺99]), wohingegen im allgemeinen Fall die Berechnung einer Zeichnung mit minimaler Fläche NP-schwer ist ([Pat99]).

Es existieren verschiedene Varianten für Gitterzeichnungen von Graphen, die sich darin unterscheiden, wie mit Knoten vom Grad größer vier verfahren wird. Wir werden darauf genauer in Kapitel 3 eingehen.

Wir wenden uns nun speziellen Definitionen zu, die in direktem Zusammenhang mit dem hier entwickelten Verfahren stehen. Sie beziehen sich vor allem auf das Verhalten der Zeichnung eines Graphen im interaktiven Umfeld.

2.2 Orthogonale Graphen im interaktiven Kontext

Wir werden im Rahmen dieser Arbeit ein *interaktives Zeichenverfahren* entwickeln, bei dem ein Benutzer durch eine Sequenz von Operationen den Graphen Schritt für Schritt erzeugt. In der Literatur wird auch der Begriff *dynamisch* verwendet. Oft ist es auch erlaubt, dass bestimmte Knoten oder Kanten wieder gelöscht werden. In diesem Fall spricht man auch von einem *voll-dynamischen Zeichenverfahren*. Im Gegensatz dazu ist in einem *semidynamischen Verfahren* nur das Einfügen von Objekten erlaubt.

Manchmal wird in der Literatur auch der Begriff *inkrementell* für interaktive Zeichenverfahren verwendet. Bei den meisten Autoren wird aber ein Algorithmus dann als inkrementell bezeichnet, wenn ein Graph gegeben ist, für diesen Graphen eine Reihenfolge der Knoten bestimmt wird und die Zeichnung des Graphen dann Knoten für Knoten mit der davor festgelegten Reihenfolge berechnet wird. Im Gegensatz zu interaktiven Algorithmen wird hier die Reihenfolge, in der die Knoten in die Zeichnung eingefügt werden, vom Algorithmus und nicht vom Anwender bestimmt. Im Folgenden wollen wir eine Notation für die Änderungen am Graphen und seiner Zeichnung in einem interaktiven Kontext entwickeln.

In einem interaktiven Zeichenverfahren wird der Graph mit Hilfe einer Menge von Operationen Schritt für Schritt aufgebaut. Wir bezeichnen den Graphen $G = (V, E)$ nach der k -ten Operation mit $G^{(k)} = (V^{(k)}, E^{(k)})$.⁴ $G^{(0)} = (V^{(0)}, E^{(0)})$ ist der Anfangsgraph. Eine Operation op auf dem Graphen $G = (V, E)$ bewirkt eine Änderung seiner Struktur, also von V , E oder beiden. Bei einer Sequenz von Operationen beschreibt $op^{(k)}$ die k -te Operation. Es gilt also mit $G^{(k-1)} = (V^{(k-1)}, E^{(k-1)})$, dass $G^{(k)} = (V^{(k)}, E^{(k)}) = op^{(k)}(G^{(k-1)})$ ist. Ein Spezialfall ist $op^{(0)}$, denn hier existiert kein Ausgangsgraph, also $G^{(0)} = (V^{(0)}, E^{(0)}) = op^{(0)}()$. Wir verwenden $v^{(k)}$ bzw. $e^{(k)}$ zur Beschreibung eines Knotens v oder einer Kante e des Graphen nach der k -ten Operation, falls der Knoten oder

⁴ Wir haben diese Notation mit der Hochstellung gewählt, weil ein Index, also z.B. v_1 in der Regel zur Nummerierung mehrerer Knoten und Kanten verwendet wird.

die Kante im Graph $G^{(k)}$ existiert.

Im Laufe der Arbeit lassen wir die Angabe, nach welcher Operation ein Objekt des Graphen betrachtet wird, wenn möglich auch weg. Wir schreiben dann z. B. wieder $G = (V, E)$ statt $G^{(k)} = (V^{(k)}, E^{(k)})$.

Wir werden nun die Operationen definieren, mit denen wir uns vor allem beschäftigen.

- $G^{(0)} = (V^{(0)}, E^{(0)})$: $G^{(0)}$ ist ein initialer Graph gemeinsam mit seiner Einbettung. Der initiale Graph ist bei uns der vollständige Graph, bestehend aus drei Knoten.
- Mit $G^{(i)} = (V^{(i)}, E^{(i)})$ gilt: $G^{(i+1)} = (V^{(i+1)}, E^{(i+1)}) = \text{InsertEdge}(v_1, v_2, f, G^{(i)})$: Wir fordern, dass f eine Fläche von $G^{(i)}$ ist und dass v_1 und v_2 auf dem Rand der Fläche f liegen. Der Parameter f gibt die Fläche an, in der die neue Kante liegen soll. Die Fläche f darf dabei nicht die Außenfläche sein. Wir können mit dieser Funktion also nur eine Kante zwischen zwei Knoten einfügen, wenn die Knoten in der selben Fläche liegen.

Es gilt $V^{(i+1)} = V^{(i)}$, $E^{(i+1)} = E^{(i)} \cup \{v_1, v_2\}$. Die neue Kante spaltet die Fläche f in zwei neue Flächen f_1 und f_2 . Abbildung 2.2 illustriert das Einfügen einer Kante.

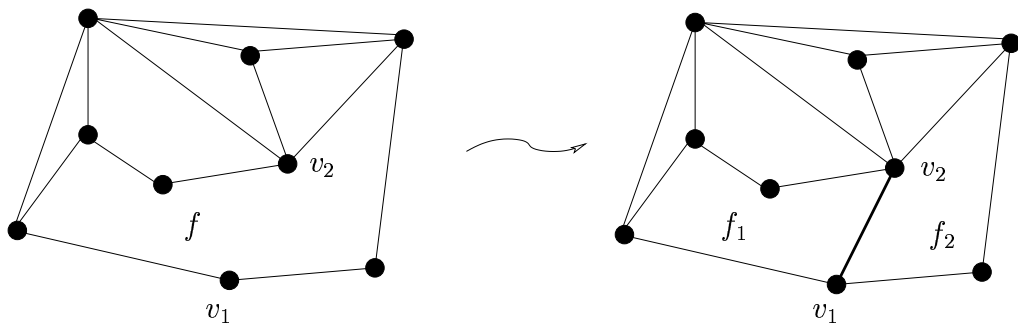


Abbildung 2.2: Einfügen einer Kante.

- Mit $G^{(i)} = (V^{(i)}, E^{(i)})$ gilt: $G^{(i+1)} = (V^{(i+1)}, E^{(i+1)}) = \text{InsertNode}(e, G^{(i)})$: Der neue Knoten v wird auf der Kante $e = \{v_1, v_2\} \in E^{(i)}$ eingefügt. Dadurch wird die Kante e durch die beiden neuen Kanten $\{v_1, v\}$ und $\{v, v_2\}$ und den Knoten v ersetzt.

Es gilt also, dass $V^{(i+1)} = V^{(i)} \cup v$ und $E^{(i+1)} = E^{(i)} \setminus e \cup \{v_1, v\} \cup \{v, v_2\}$. In den beiden an e angrenzenden Flächen wird e durch die beiden neuen Kanten und den neuen Knoten ersetzt. Abbildung 2.3 illustriert das Einfügen eines Knotens.

Die Mächtigkeit dieser Operationen zeigt folgender Satz.

Satz 1

Mit einer Initialisierungsfunktion und diesen Operationen kann jeder zweifach zusammenhängende planare Graph erzeugt werden ([CBTT95]).

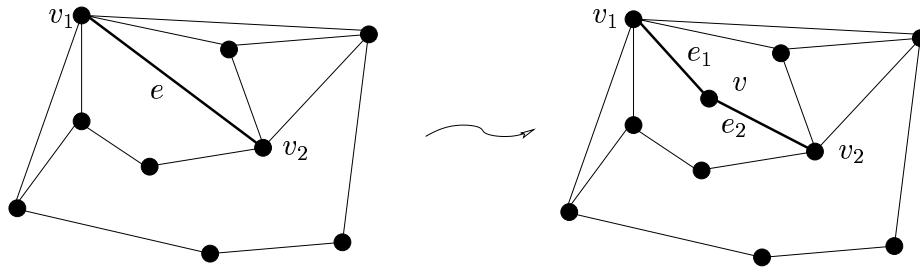


Abbildung 2.3: Einfügen eines Knotens.

Die Operationen beschreiben nur die Änderungen am Graphen. Wenn eine Zeichnung existiert, dann muss sie ebenfalls geändert werden — es muss mindestens ein neues Objekt eingefügt oder ein altes gelöscht werden — um die Modifikation am Graphen darzustellen. Die entsprechenden Anpassungen an der Zeichnung werden im Rahmen der vorliegenden Arbeit festgelegt.

Nachdem nun alle Voraussetzungen geschaffen sind, wenden wir uns den aus der Literatur bekannten Zeichenverfahren zu. Wir beschränken uns dabei auf die für uns wichtigsten orthogonalen und dynamischen Zeichenmodelle und Methoden.

3 Orthogonale und interaktive Zeichenmodelle und -verfahren

In diesem Kapitel wollen wir einen Überblick über bekannte Algorithmen zum Zeichnen von Graphen geben, die in Zusammenhang mit dem Thema dieser Arbeit stehen. Dazu beschreiben wir zuerst die verschiedenen orthogonalen Zeichenmodelle, gemeinsam mit den wichtigsten Zeichenverfahren für das jeweilige Modell. Anschließend beschäftigen wir uns mit interaktiven Algorithmen. Die Ergebnisse der hier vorgestellten Arbeiten werden in den späteren Kapiteln mit den Ergebnissen aus dieser Arbeit verglichen.

3.1 Statische orthogonale Zeichenmodelle

Es existiert eine Vielzahl von Algorithmen zum statischen orthogonalen Zeichnen von Graphen, die sich im Hinblick auf das verwendete Zeichenmodell, der algorithmischen Herangehensweisen und der betrachteten Graphklassen unterscheiden. In [Bie95] werden einige dieser Verfahren gegenübergestellt und bzgl. Laufzeit, Knickzahl und Fläche verglichen. In [BETT98] werden neben anderen Algorithmen auch diverse Algorithmen zum orthogonalen Zeichnen von Graphen beschrieben. Wir verzichten in dieser Arbeit auf eine genauere Analyse der statischen Zeichenverfahren, da wir uns mit interaktiven Verfahren beschäftigen und Vergleiche zwischen statischen und interaktiven Zeichenverfahren nur schwer möglich sind. Statt dessen konzentrieren wir uns auf die zu Grunde liegenden Zeichenmodelle.

Eine Gemeinsamkeit der Verfahren besteht darin, dass die Objekte des Graphen auf einem zugrundeliegenden Gitter platziert werden. Es handelt sich also um Gitterzeichnungen. Neben den verschiedenen Zeichenmodellen können einige Verfahren auch Einschränkungen der erlaubten Zeichnungen, wie beispielsweise vordefinierte Lagebeziehungen zwischen Knoten, berücksichtigen.

Wir werden im Folgenden die Zeichenmodelle und die wichtigsten Arbeiten je Modell beschreiben. In der hier vorliegenden Arbeit wird das Kandinsky-Modell verwendet, das in Abschnitt 3.1.4 betrachtet wird.

3.1.1 Punktförmige Knoten

Bei diesem Zeichenmodell liegen die Knoten auf den Schnittpunkten von Gitterlinien und haben keine Ausdehnung. Unter der Voraussetzung, dass Kanten auf Gitterlinien

platziert werden und sich nicht überlagern dürfen, führt das dazu, dass die Knoten gradbeschränkt sind. An jeder Seite eines Knotens kann nur eine Kante angeschlossen werden und damit beträgt der Grad eines Knotens maximal vier. Diese Beschränkung auf den Grad 4 ist auch das Hauptproblem dieses Ansatzes, weil das bei praxisnahen Graphen normalerweise nicht gegeben ist. Abbildung 3.1 zeigt ein Beispiel dieses Knotenmodells.

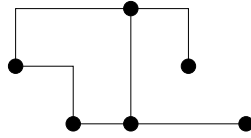


Abbildung 3.1: Beispiel einer Zeichnung im punktförmigen Zeichenmodell.

Eine der ersten Arbeiten im Umfeld von Graph Drawing in diesem Modell¹ stammt von Tamassia [Tam87]. Dabei ist ein eingebetteter Graph mit maximalem Grad vier gegeben und es wird eine knickminimale Zeichnung berechnet. Der Algorithmus von Tamassia arbeitet in zwei Schritten: Im ersten wird eine orthogonale Repräsentation berechnet, für die dann in einem zweiten Schritt eine Zeichnung generiert wird.

Die Optimierung der Anzahl der Kantenknicke wird mit Hilfe einer Transformation in ein Netzwerkflussproblem gelöst. Die Problem Instanz für die Berechnung des Flusses mit minimalen Kosten wird dabei so definiert, dass einem Fluss mit Kosten k eine Zeichnung mit k Knicken entspricht. Die sich aus dem Flussminimierungsalgorithmus ergebende Laufzeit von $O(n^2 \cdot \log n)$ konnte in [GT96] durch die Entwicklung eines speziellen Algorithmus zur Flussminimierung auf $O(n^{7/4} \cdot \sqrt{\log n})$ verbessert werden.

Für nicht eingebettete, planare Graphen wird in [GT94] gezeigt, dass die Berechnung einer planaren Zeichnung mit der minimalen Knickzahl im punktförmigen Knotenmodell NP-vollständig ist. In [DL97] wird die exponentielle Implementierung eines „branch and bound“ Algorithmus vorgestellt, der auch für nicht eingebettete Graphen eine knickminimale Zeichnung berechnet. Dabei hat der Benutzer die Möglichkeit, in den „branch and bound“ Algorithmus einzugreifen und das Programm auf diese Art zu unterstützen.

Bei der Berechnung einer Zeichnung aus einer orthogonalen Repräsentation wird in [Tam87] eine einfache Heuristik verwendet. Jede Fläche wird in eine Menge von Rechtecken partitioniert und für dieses System von Rechtecken kann dann eine planare Zeichnung erzeugt werden.

Eine interessante Frage bei diesem Schritt ist es, für die gegebene orthogonale Repräsentation eine flächenminimale Zeichnung zu finden. Patrignani ([Pat99]) zeigt, dass dieses Problem NP-vollständig ist. In dieser Arbeit wird außerdem gezeigt, dass sowohl die Minimierung der längsten Kante als auch die Minimierung der Summe der Kantenlängen NP-vollständig sind. Für die eingeschränkte Klasse von orthogonalen Repräsentationen, die „turn-regular“ sind, wird in [BBD⁺99] ein optimaler Linearzeitalgorithmus

¹ Im Umfeld von VLSI-Layout hat man sich schon vorher mit der Platzierung und Objekten und ihrer Verdrahtung beschäftigt.

vorgestellt. Auch für beliebige orthogonale Repräsentationen existiert ein (exponentieller) Algorithmus zur Berechnung einer Zeichnung ([KM99b]), bei dem für das Problem einer Zeichnung mit minimaler Summe der Kantenlängen eine ILP-Formulierung angegeben wird. Nach Aussage der Autoren hat die Lösung dieses ILP auch für große Graphen eine akzeptable Laufzeit.

Andere Algorithmen für das punktförmige Zeichenmodell basieren auf einem inkrementellen Ansatz (siehe z. B. [PT94, Bie95, PT96]). Dabei wird für einen gegebenen (nicht für alle Algorithmen notwendigerweise planaren) Graphen eine Ordnung der Knoten berechnet, nach der die Zeichnung des Graphen Knoten für Knoten aufgebaut wird. Beim Platzieren eines Knotens wird unter Berücksichtigung der Lage der schon platzierten Nachbarn seine Position und der Kantenverlauf zu den Nachbarn festgelegt. Diese Verfahren kombinieren eine geringe, im Normalfall lineare Laufzeit mit kleiner Knickzahl und kleiner Fläche, wobei sowohl bei der Fläche als auch bei der Knickzahl in der Regel nicht das Optimum erreicht wird.

Um die Beschränkung des Knotengrades auf vier zu umgehen, existiert als eine Variante von diesem Modell das „quasi-orthogonale Zeichnen“ (vgl. [Kan96, KM98, GM98]). Hier ist es erlaubt, dass bei einem Knoten mit Grad größer vier das erste Segment einer Kante diagonal bis zu einem Gitterpunkt verläuft. Bild 3.2 zeigt eine Zeichnung in diesem Modell. In diesem Modell können damit alle Graphen (auch nicht planare) dargestellt werden, was eine große Verbesserung im Vergleich zum ursprünglichen punktförmigen Knotenmodell bedeutet.

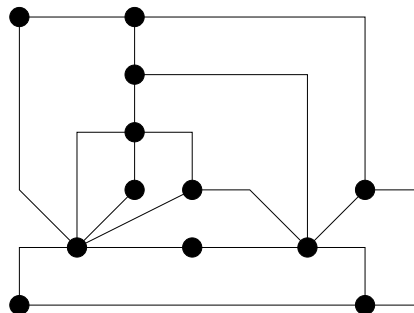
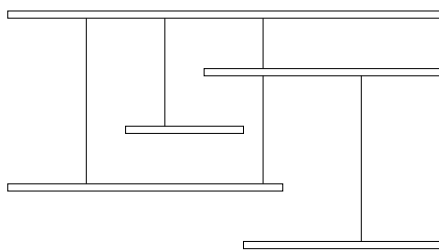


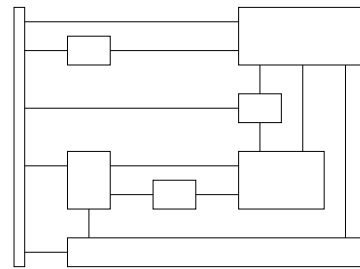
Abbildung 3.2: Beispiel einer quasi-orthogonalen Zeichnung im punktförmigen Zeichenmodell.

3.1.2 Zeichenmodell für unbeschränkte Knotengrößen

In diesem Modell dürfen Knoten eine beliebige Größe besitzen. Damit ist es möglich, planare Graphen ohne Kantenknicke zu zeichnen (siehe z. B. [RT86, TT86, Kan97]). Solche Zeichnungen werden (schwache) Sichtbarkeitsdarstellungen (engl. „visibility representation“) genannt. Dabei werden Knoten als waagerechte Boxen und Kanten als vertikale Linien dargestellt. Bild 3.3(a) zeigt ein Beispiel für eine Sichtbarkeitsdarstellung.



(a) Beispiel einer Sichtbarkeitsdarstellung.



(b) Beispiel einer 2-Sichtbarkeitsdarstellung.

Abbildung 3.3: Beispiele für Sichtbarkeitsdarstellungen.

Daneben gibt es Variationen von diesem klassischen Modell. Dabei wird versucht, mit großen Knoten die Anzahl an Kantenknicken gegenüber Algorithmen mit kleinen Knoten zu verringern. In [FKK96] wird beispielsweise das Konzept von 2-Sichtbarkeit (engl. „2-visibility“) eingeführt. Knoten sind in diesem Modell keine vertikalen Boxen, sondern ihre Ausdehnung kann sowohl in x- als auch in y-Richtung variiert werden. Kanten dürfen sowohl horizontal als auch vertikal verlaufen. Damit können nun nicht mehr nur planare Graphen, sondern auch nicht planare Graphen wie z.B. der K_5 dargestellt werden, wenn man erlaubt, dass sich waage- und senkrechte Kanten schneiden. Bild 3.3(b) zeigt ein Beispiel für eine 2-Sichtbarkeitsdarstellung. Für dieses Zeichenmodell wird in [FKK96] ein Algorithmus für Zeichnungen ohne Kantenknicke angegeben. Zusätzlich wird ein weiterer Algorithmus erläutert, bei dem Knoten zur Vermeidung von Kantenknicken vergrößert werden, aber trotzdem Kantenknicke existieren dürfen. Dabei wird so vorgegangen, dass bei jedem Knoten überprüft wird, ob eine Vergrößerung eines Knotens einen Kantenknick überdecken würde. Wenn ja, dann wird der Knoten vergrößert. Bei diesem Verfahren wird also ein Kompromiss zwischen vergrößerter Knotenfläche und der Anzahl an Kantenknicken angestrebt.

Ein weiteres Konzept, das zwar beliebig große Knoten erlaubt, sich aber wesentlich von Sichtbarkeitsdarstellungen unterscheidet, ist das GIOTTO-Konzept. In [TdBB88] wird die Arbeit aus [Tam87] erweitert und dieses Konzept eingeführt. Um auch Knoten mit Grad größer vier verarbeiten zu können, werden Knoten mit größerem Grad zuerst gesplittet, so dass wieder ein Grad-4 Graph entsteht. Der Algorithmus aus [Tam87] wird so angepasst, dass auch gesplittete Knoten in der endgültigen Zeichnung wieder als Knoten gezeichnet werden, wobei allerdings die Knoten beliebig groß werden können. Bei Sichtbarkeitsdarstellungen sind beliebig große Knoten also ein wesentlicher Bestandteil des Algorithmus. Hier sind die großen Knoten die Folge der Anpassung eines schon bekannten Algorithmus an eine neue Fragestellung, die den Nachteil hat, dass man die Kontrolle über die Größe der Knoten verliert.

Durch den Einbau einer Planarisierungsphase und die Berechnung einer Einbettung

ist es möglich, für nicht planare, bzw. nicht eingebettete Graphen eine Zeichnung zu berechnen. Wie schon erwähnt können durch eine Vergrößerung der Knoten auch Graphen mit Knotengrad größer vier verarbeitet werden. Allerdings verliert man bei diesem Verfahren die Kontrolle über die Knotengröße. Die Ausdehnung eines Knotens ist nicht mehr beschränkt, weder durch seinen Grad noch durch eine Konstante. Man befindet sich also im Modell mit unbeschränkten Knotengrößen.

3.1.3 TSS-Modell

Im TSS-Modell ([Bie97]) richtet sich die Höhe und Breite eines Knotens nach der Anzahl an Kanten, die auf einer Seite eines Knotens angeschlossen werden müssen. Knoten dürfen sich in diesem Modell über mehrere Schnittpunkte von Gitterlinien erstrecken, allerdings nur über genau so viele, wie zum Anschluss der Kanten unbedingt nötig sind. Falls beispielsweise auf der Unterseite eines Knotens fünf und auf der Oberseite vier Kanten enden, dann erstreckt sich der Knoten über fünf vertikale Gitterlinien. Entsprechendes gilt für die Höhe eines Knotens. Kanten verlaufen wieder auf den Gitterlinien. Abbildung 3.4 zeigt eine Zeichnung in diesem Modell.

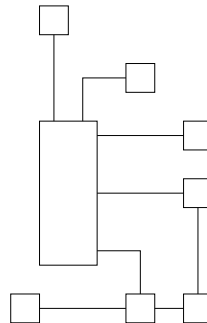


Abbildung 3.4: Beispiel einer Zeichnung im TSS-Modell.

Algorithmen für das TSS-Modell werden vor allem in der Dissertation von Biedl [Bie97] und in [BMT97] vorgestellt. Sie beschäftigen sich dabei mit planaren und nicht planaren Graphen. Außerdem werden Einschränkungen der Zeichnungen durch die Vorgabe von Lagebeziehungen oder vordefinierten, minimalen Abmessungen der Knoten betrachtet.

In diesem Modell können also Knoten mit beliebigem Grad gezeichnet werden. Die Knoten haben keine fest vorgegebene Größe, allerdings ist die Größe der Knoten durch den Eingangsgrad je Knotenseite beschränkt. Bedingt durch die verschiedenen Knotengrößen können an einer Seite eines Knotens mehrere Kanten ohne Knick enden.

3.1.4 Kandinsky-Modell

Das Kandinsky-Modell wurde 1995 von Fößmeier und Kaufmann (siehe [FK95]) entwickelt.² Dieses Zeichenmodell wird detailliert erläutert, da die vorliegende Arbeit darauf basiert.

Hauptziel des Kandinsky-Modells ist eine, gegenüber dem GIOTTO-Modell verbesserte Verarbeitung von Knoten mit Grad größer als vier. Im GIOTTO-Modell führt ein großer Knotengrad zu einer unkontrollierten Vergrößerung der Knoten. Im Kandinsky-Modell darf die Größe eines Knotens nicht beliebig wachsen.

Im Gegensatz zum TSS-Modell besitzen die Knoten des ursprünglichen Kandinsky-Modells³ eine uniforme Größe. Um trotz der uniformen Knotengröße auf jeder Seite eines Knotens beliebig viele Kanten anschließen zu können, existieren im Kandinsky-Modell zwei verschiedene Gitter. Ein grobes, auf dessen Schnittpunkten sich die Knotenmittelpunkte befinden und ein feines, auf dem die Kanten verlaufen. Die feinen Gitterlinien verlaufen parallel zu den groben und liegen so nahe zusammen, dass sie die Knoten schneiden. Man kann damit eine Menge von feinen Gitterlinien als Ausprägungen einer groben Gitterlinie betrachten. Der Abstand zwischen je zwei feinen Gitterlinien wird so klein gewählt, dass auf jeder Seite eines Knotens so viele Kanten angeschlossen werden können wie nötig. In Abbildung 3.5 sind die Gitter schematisch dargestellt.

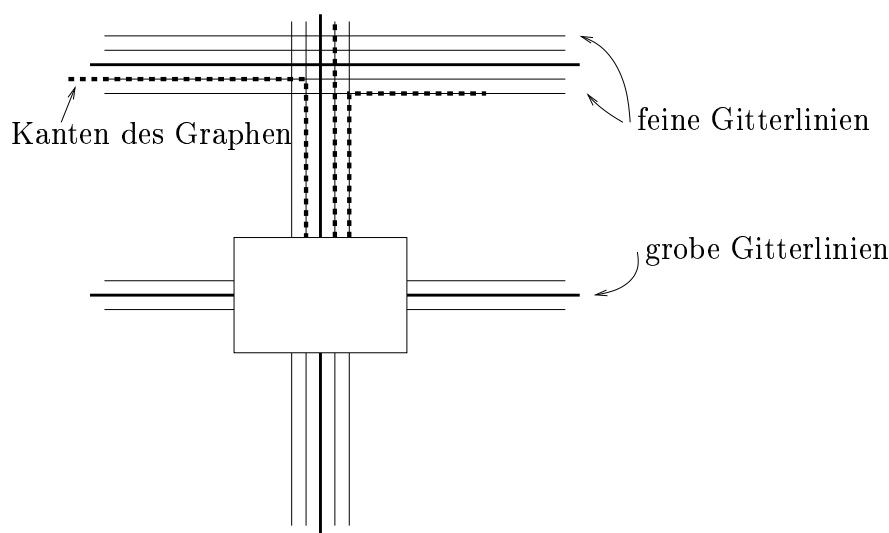


Abbildung 3.5: Feine und grobe Gitterlinien im Kandinsky-Modell.

Auf Grund der uniformen Knotengröße kann im Gegensatz zum TSS-Modell bei einer Zeichnung ohne Knoten-Kanten-Schnitte auf jeder Seite eines Knotens nur eine Kante geradlinig angeschlossen werden. In einer planaren Zeichnung müssen dann alle anderen Kanten noch vor dem Endknoten der Kante ohne Knick jeweils nach links oder rechts

² Das Zeichenmodell wird in dieser Arbeit erstmals beschrieben. Der Name Kandinsky-Modell wurde allerdings erst später vergeben.

³ Es gibt Varianten, in denen die Knotengröße variieren kann.

abknicken. Diese Eigenschaft von planaren Zeichnungen im Kandinsky-Modell wird als „bend-or-end“ Eigenschaft bezeichnet. Ein Kante ohne Knicke verläuft dabei immer von Knotenmittelpunkt zu Knotenmittelpunkt, also auf einer groben Gitterlinie.

In [FK95] wird unter anderem der Algorithmus von [Tam87] so abgeändert, dass nun planare, eingebettete Graphen mit Grad größer vier verarbeitet werden können und Zeichnungen im Kandinsky-Modell erzeugt werden. Die Zeichnungen sind wieder knickminimal, allerdings nur unter der Voraussetzung, dass Flächen, die durch mindestens drei Knoten begrenzt werden, nicht „leer“ gezeichnet werden dürfen. In Abbildung 3.6 ist eine leere Fläche dargestellt.

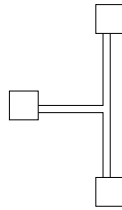


Abbildung 3.6: Bei der Knickminimierung im Kandinsky-Modell verbotene Flächenform.

In [FK97] wird eine weitere Variante von diesem Algorithmus vorgestellt, bei der nicht planare Graphen verarbeitet werden können. Diese Variante baut nicht auf der üblichen Strategie auf, zuerst den Graphen zu planarisieren und dann den Standardalgorithmus zu verwenden, sondern verarbeitet den nicht-planaren Graphen direkt. Eine Planarisierung würde Dummyknoten auf den Kantenschnitten voraussetzen, was zu einer unnötig hohen Zahl an Kantenknicken führen würde(vgl. [FK97]).

Hier wird außerdem eine Variante des Kandinsky-Modells, das „Kandinsky-Bignode-Modell“ eingeführt. Um die benötigte Fläche einer Zeichnung besser beurteilen zu können, werden hier die feinen Gitterlinien in grobe umgewandelt. Damit verliert man die Uniformität der Knotengrößen und befindet sich im TSS-Modell.

In [EFK00] wird eine weitere Modellvariante vorgestellt, bei der das Problem der Knickminimierung für planare, eingebettete Graphen als ILP⁴ formuliert wird. Dabei bleibt die Laufzeit laut Autoren klein und man gewinnt zusätzliche Flexibilität, um z. B. Einschränkungen der Zeichnungen durch vorgegebene Lagebeziehungen, in der Eingabe festgelegte Knotengrößen oder Kantenbeschriftungen mit berücksichtigen zu können. Wenn Knotengrößen Teil der Eingabe sind, dann genügt das „Kandinsky-Bignode-Modell“ nicht, da in diesem Modell die Größe eines Knotens von seinem Grad abhängt. Hierfür dient die neue Variante dieses Modells, das „Relaxierte Kandinsky-Modell“, das in [Eig99] genauer beschrieben ist.

Wir haben uns für das Kandinsky-Modell und gegen das TSS-Modell entschieden, weil im Kandinsky-Modell die Größe eines Knotens unabhängig vom Grad des Knotens ist.

⁴ Eine ILP Formulierung eines Problems beschreibt dieses als ein ganzzahliges System von Ungleichungen. Für dieses System ist eine Zielfunktion vorgegeben, die innerhalb des Systems der Ungleichungen minimiert werden muss.

Unserer Meinung nach ist es nicht im Sinne des Anwenders, wenn die Ausdehnung eines Knotens während einer interaktiven Sitzung durch das Programm geändert wird. Dazu wäre es nicht einmal nötig, dass sich der Grad eines Knotens ändert. Es genügt, wenn die Anschlussstellen der Kanten auf den verschiedenen Knotenseiten neu verteilt werden. In einem interaktiven Algorithmus soll sich aber die Zeichnung nur so viel wie unbedingt nötig ändern, was auch für die Knotengrößen gilt.

Davon abgesehen gibt es nur noch einen weiteren Unterschied zwischen den beiden Zeichenmodellen: Im TSS-Modell ist es erlaubt, dass auf einer Seite eines Knotens mehrere Kanten ohne Knick enden. Im Kandinsky-Modell ist das wegen der „bend-or-end“-Eigenschaft unmöglich. Allerdings erscheint uns dieser Vorteil des TSS-Modells als zu gering im Vergleich zur vom Grad unabhängigen Knotengröße, so dass wir nicht das TSS-Modell verwenden.

Neben diesen Zeichenmodellen und den hier beschriebenen Algorithmen existieren eine Vielzahl von weiteren Algorithmen, die sich zum Teil mit eingeschränkten Graphklassen oder mit speziellen Zeichenstilen beschäftigen.

Hier gibt es zum Beispiel eine orthogonale Variante von hierarchischen Zeichnungen ([San95]) oder planares, geclustertes Zeichnen ([EF97]). Für Bäume existiert ein Verfahren zum geradlinigen orthogonalen⁵ Zeichnen mit einstellbarem Verhältnis von Höhe zu Breite der Zeichnung. In letzter Zeit werden zunehmend auch Algorithmen für 3-dimensionales orthogonales Zeichnen entwickelt (siehe z. B. [ESW96, BSWW97, Bie98, BPV98, Woo98]). Wir gehen hier auf diese Verfahren nicht weiter ein, weil sie in keinem Zusammenhang zu der vorliegenden Arbeit stehen.

3.2 Interaktive Zeichenverfahren

Ein interaktiver Algorithmus zeichnet sich dadurch aus, dass es keinen fertigen Graphen als Eingabe für den Algorithmus gibt. Statt dessen verändert ein Benutzer den Graphen im Rahmen einer interaktiven Sitzung am Computer Schritt für Schritt. Nach jeder Veränderung⁶ wird eine neue Zeichnung erstellt.

Dabei unterscheiden sich die verschiedenen Algorithmen zunächst nach der Menge an Operationen, mit denen der Graph manipuliert werden darf. Die wichtigsten Operationen zum Erzeugen von Knoten und Kanten sowie zum Löschen von einzelnen Objekten sind:

1. Erzeugung eines initialen Graphen.
2. Erzeugung eines neuen, isolierten Knotens.
3. Erzeugen eines neuen Knotens durch das Teilen einer Kante. Dabei wird die Kante $\{v_1, v_2\}$ durch einen neuen Knoten v und die Kanten $\{v_1, v\}$ und $\{v, v_2\}$ ersetzt, so dass ein neuer Knoten entsteht.
4. Erzeugung einer neuen Kante zwischen zwei vorhandenen Knoten.

⁵ Geradlinig orthogonal heißt, dass eine Kante orthogonal verläuft und keine Knicke hat.

⁶ Manchmal wird die Zeichnung auch erst nach mehreren Veränderungen erstellt.

5. Als Spezialfall davon bei planaren Graphen: Erzeugung einer neuen Kante zwischen zwei vorhandenen Knoten in einer gemeinsamen Fläche.
6. Erzeugung eines neuen Knotens, gemeinsam mit einer Menge von Kanten zu einer Auswahl von schon vorhandenen Knoten.
7. Löschen eines Knotens und aller adjazenten Kanten.
8. Löschen einer Kante.
9. Löschen von Subgraphen mit bestimmten Eigenschaften (wobei dies auch durch eine Folge von elementaren Löschoptionen für einzelne Knoten und Kanten gelöst werden kann).

Verschiedene Algorithmen nutzen die unterschiedlichsten Auswahlen aus dieser Menge zum Erzeugen der Graphen. Die Operationen drei und fünf gemeinsam mit einer Initialisierungsfunktion, die einen vollständigen Graphen mit drei Knoten erzeugt, genügen, um jeden planaren zweifach zusammenhängenden Graphen zu erzeugen. Um die Menge aller Graphen zu erzeugen, kann z.B. eine Kombination der Operationen eins, zwei und vier verwendet werden oder alleine die Funktion sechs.

Neben reinen Zeichenalgorithmen gibt es auch noch eine Reihe von Datenstrukturen wie z.B. SPQR-Bäume ([BT96]) und zugehörigen Algorithmen, die jeweils für eine bestimmte Untermenge der obigen Operationen bestimmte Eigenschaften des Graphen überprüfen und verwalten. Diese Datenstrukturen dienen oft als Basis für Zeichenalgorithmen, die die Informationen über die Eigenschaften des Graphen zur Berechnung der Zeichnung benötigen.

Bei interaktiven Zeichenalgorithmen muss im Gegensatz zu statischen Algorithmen beachtet werden, dass neben dem Graphen auch eine Zeichnung des Graphen existiert. Die Eingabe für einen Zeichenalgorithmus besteht hier aus dem Graphen, seiner Zeichnung und einer Operation zur Veränderung der Struktur des Graphen. Die Änderungen an der Struktur des Graphen sind dabei normalerweise lokal und klein im Gegensatz zum Gesamtgraphen. Die Anpassung der vorhandenen Zeichnung muss dann zwei im Allgemeinen widersprüchliche Anforderungen erfüllen: Die generierte Zeichnung soll bestimmten Ästhetikkriterien bestmöglich entsprechen und außerdem sollen die Änderungen an der Zeichnung gering sein, denn eine große Änderung der Zeichnung zwingt den Benutzer, sich vollständig neu zu orientieren. Dieser Aspekt wird als Erhaltung der „mental map“ bezeichnet. Der Begriff wurde von Misue et al. in [MELS95] geprägt.

Die Autoren stellen drei verschiedene Modelle zur qualitativen Beschreibung der „mental map“ mittels einer Funktion vor:

- Orthogonale Ordnung: Die relativen Lagebeziehungen von je zwei Knoten sollen bestmöglich erhalten bleiben. Das heißt, es wird gezählt, wie viele Knotenpaare die Reihenfolge in x- und y-Richtung in einer neuen Zeichnung tauschen.

- Nächste Nachbarn: Man berechnet einen Graphen zur Charakterisierung der geometrischen Nachbarschaften. Diese sollen bestmöglich erhalten bleiben. Nach der Neuberechnung bewertet man die Änderungen an diesem Graph.
- Topologie: Eine generelle Anforderung an Zeichenalgorithmen soll es sein, dass die Einbettung des Graphen nicht verändert werden darf, außer an der Stelle, die durch eine Operation auf dem Graphen verändert wurde.

In [BT98] werden die Ideen aus [MELS95] verfeinert und erweitert. Es werden zunächst vor der Berechnung der Differenz der beiden Zeichnungen noch geometrische Anpassungen vorgenommen, die Verfälschungen durch unterschiedliche Skalierungen, Verschiebungen und Rotationen vermeiden sollen.

Zur Berechnung der Unterschiede von zwei verschiedenen Zeichnungen wird aus den Bildern eine Menge von Punkten gewonnen. Dies sind im Normalfall die Knotenmittelpunkte und die Kantenknicke. Wenn auch Änderungen der Knotenform und Größe mit berücksichtigt werden sollen, dann können anstatt den Knotenmittelpunkten auch die Eckpunkte der Knoten verwendet werden. Nun werden verschiedene Metriken festgelegt, die den Abstand zwischen je zwei Punktemengen bestimmen. In der Arbeit werden Metriken zur Bewertung

- des geometrischen Abstands zwischen der alten und neuen Position eines Punktes,
- der Erhaltung der Nachbarschaften,
- der Beibehaltung der orthogonalen Ordnung und
- der Konservierung der Kantenverläufe

vorgestellt.

Im Bereich der interaktiven Zeichenalgorithmen gibt es zunächst (nicht orthogonale) Ansätze für eine große Menge von eingeschränkten Graphklassen.

In [Nor95] wird eine interaktive Variante eines DAG-Zeichenalgorithmus vorgestellt. Hier werden die interaktiven Anforderungen an den Algorithmus in topologische — also die Einbettung — und geometrische — z. B. die relative Lage von Knoten — Aspekte geteilt. Diese sollen bestmöglich erhalten werden. Allerdings wird nicht angegeben, wie diese Aspekte erreicht werden und wie gut sie erreicht werden.

In [CBTT95, CBT⁺92] beschäftigen sich die Autoren mit interaktivem Zeichnen von Bäumen, planaren gerichteten Serien-Parallel-Graphen, planaren st-Graphen und zweifach zusammenhängenden planaren Graphen. Die Invarianten jedes Verfahrens setzen sich aus statischen und dynamischen Komponenten zusammen, die vom Algorithmus eingehalten werden müssen. Die statische Komponente bewertet dabei die Qualität der Zeichnung und die dynamische die Ähnlichkeit zwischen den verschiedenen Zeichnungen vor und nach einer Änderung der Graphstruktur durch eine Operation. Neben Operationen zur Änderung des Graphen gibt es noch Anfrageoperationen zur Berechnung einer

neuen Zeichnung. Je nach Graphklasse betrachten sie verschiedene Operationen und verschiedene Komponenten der Invarianten. Die Algorithmen für alle Klassen beruhen dabei auf der gemeinsamen Idee, geeignete Teilgraphen durch geometrische Objekte zu repräsentieren. Der Teilgraph wird dann innerhalb dieses geometrischen Objekts gezeichnet. Bei Bäumen sind das z. B. Rechtecke, die den jeweiligen Subbaum ab einem bestimmten inneren Knoten repräsentieren. Der Subbaum muss dann innerhalb des Rechteckes gezeichnet werden.

Da auch wir uns mit planaren zweifach zusammenhängenden Graphen und den gleichen Einfügeoperationen beschäftigen, soll dieser Algorithmus genauer betrachtet werden. Für zweifach zusammenhängende planare Graphen geben sie einen semi-dynamischen Algorithmus mit den Operationen „Einfügen eines Knotens auf einer Kante“ und „Einfügen einer Kante in einer Fläche“ an, bei dem die Einfügeoperationen und das Zeichnen in $O(\log n)$ möglich sind. Das Zeichnen bezieht sich dabei auf das Zeichnen eines einzelnen Knoten oder einer einzelnen Kante. Das heißt, die Darstellung von t Objekten kostet $O(t \cdot \log n)$. Dabei wird der Graph intern als gerichteter planarer st-Graph verwaltet. Bei den Zeichnungen wird die planare Einbettung erhalten. Jede Kante besitzt maximal einen Knick, wobei die Zeichnungen nicht orthogonal sind. Insgesamt hat die Zeichnung maximal $2n - 5$ Kantenknicken. Da ein planarer Graph nach dem Satz von Euler maximal $3n - 6$ Kanten haben kann, sind ungefähr n Kanten geradlinig gezeichnet. Man beachte, dass außer dem Erhalt der Einbettung keine weiteren Forderungen bzgl. der Ähnlichkeit von aufeinander folgenden Zeichnungen gemacht werden. Die Invarianten zum Erhalt der „mental map“ des Anwenders sind hier also nicht so stark ausgeprägt. Das Erstellen einer Zeichnung des gesamten Graphen kostet $O(n \cdot \log n)$ Laufzeit.

In [TBC97] wird eine semi-dynamische Variante eines Springembedders zum geradlinigen Zeichnen von Graphen verwendet, um die Ergebnisse einer Dokumentensuche graphisch darzustellen. Hierbei genügt ein semi-dynamisches Verfahren, da der jeweils neue Graph auf einer Anfrageverfeinerung beruht und damit immer nur neue Objekte hinzukommen.

In [BW97] zeigen Brandes und Wagner, wie allgemein für einen gegebenen, statischen Algorithmus eine dynamische Variante entwickelt werden kann. Diese Arbeit basiert auf dem „random field model“. In diesem Modell wird jeder Zeichnung eine Wahrscheinlichkeit zugeordnet, mit der sie ausgewählt wird. Die Wahrscheinlichkeit hängt von der Güte der Zeichnung ab. Damit kann dieses Modell als gemeinsame Basis für alle Zeichenalgorithmen betrachtet werden. Für dynamische Algorithmen können nun bei der Bewertung der Güte einer Zeichnung auch interaktive Komponenten mit einfließen. Man stellt sich also die Frage, wie gut eine Zeichnung ist, unter der Voraussetzung, dass die Vorgängerzeichnung eine bestimmte Form hat. Die Folge sind dann bedingte Wahrscheinlichkeiten.

Die Verwendung des Modells wird an den Beispielen eines Springembedders und einer interaktiven Variante des orthogonalen Zeichenalgorithmus von Tamassia ([Tam87]) erläutert. Die interaktive Erweiterung des Algorithmus von Tamassia erlaubt Einfüge- und Löschoptionen. Als interaktive Komponente der Bewertungsfunktion kommt noch die Änderung der Anzahl an Knicken je Kante hinzu. Der Verlauf einer Kante wird aber ignoriert und darf sich damit beliebig ändern, ohne die Bewertung der Zeichnung zu ver-

schlechtern. Die Einschränkung aus der Arbeit von Tamassia, dass jeder Knoten maximal Grad vier aufweisen darf, bleibt erhalten.

In [BW98] und [Bra99] wird das dynamische, orthogonale Zeichenverfahren aus [BW97] so erweitert, dass bei der interaktiven Komponente der Bewertungsfunktion auch eine Änderung eines Kantenverlaufes durch höhere Kosten der Folgezeichnung bestraft wird. Es werden nun also auch die Änderung der Richtung der Knicke einer Kante und die Änderung der Seite, auf der eine Kante einen Knoten verlässt, bestraft.

In [BGW00] wird schließlich zusätzlich die Einschränkung, dass ein Knoten maximal Grad vier haben darf, aufgegeben. Dies wird dadurch erreicht, dass entweder wie beim GIOTTO-Verfahren die Knoten beliebig wachsen dürfen bzw. quasiorthogonale Zeichnungen im punktförmigen Zeichenstil verwendet werden. Zur Berechnung einer neuen Zeichnung muss eine Optimierung mit Hilfe einer Transformation in ein Netzwerk-Flussproblem gelöst werden.

Man erhält hier einen Kompromiss zwischen statischen und dynamischen Anforderungen an die Zeichnung. Es gibt keine Garantie, dass bestimmte Minimalanforderungen an die maximale Änderung der Zeichnung eingehalten werden. Außerdem ist bei einem interaktiven System die Reaktionszeit des Systems kritisch. Bedingt durch den Algorithmus zur Berechnung des minimalen Flusses scheint die theoretische Komplexität jeder Update-Operation sehr hoch zu sein. Die Autoren machen allerdings keine Angaben zur Reaktionszeit.

Ein zweiter, zu den bisher vorgestellten Ideen grundverschiedener Ansatz zum orthogonalen interaktiven Zeichnen von nicht-planaren Graphen, wird in [PT95] verfolgt. Der Ansatz basiert auf dem punktförmigen Knotenmodell und damit ist der Grad eines Knotens auf vier begrenzt. In [PST96] erfolgt die praktische Untersuchung der vorgestellten Verfahren an 11491 Graphen aus praktischen Anwendungen.

In der Arbeit werden zunächst verschiedene Szenarien für das interaktive Zeichnen von Graphen vorgestellt. Im ersten Ansatz werden keine Anforderungen an die Ähnlichkeit zwischen zwei aufeinander folgenden Zeichnungen gestellt. Damit können alle Standardalgorithmen zum statischen Zeichnen verwendet werden.

Die in der Arbeit eigentlich untersuchten Szenarien heißen „no change“ und „relative coordinates“. Für beide Szenarien wird angegeben, wie je ein neuer Knoten gemeinsam mit seinen Kanten zu den schon gezeichneten Knoten eingefügt werden muss.

Bei „no change“ darf die einmal gewählte Position eines Knotens nicht mehr verändert werden. Damit können neue Knoten immer nur außen um die schon vorhandene Zeichnung angeordnet werden. Bei „relative coordinates“ muss die relative Lage zwischen je zwei Knoten erhalten bleiben. Im Gegensatz zu „no change“ können damit neue Knoten auch zwischen den schon gezeichneten Knoten eingefügt werden, da mit Hilfe von Verschiebeoperationen neue Zeilen und Spalten geschaffen werden können.

Auf Grund des gewählten Ansatzes werden planare Graphen im Allgemeinen nicht überschneidungsfrei gezeichnet.

Ein Verfeinerung dieses Ansatzes wird in [Föß97] vorgestellt. Hier erhält man eine kleinere Fläche und weniger Kantenknicke. Die besseren Ergebnisse ergeben sich aus ei-

ner geschickteren Wahl der Regeln, die festlegen, wie ein neuer Knoten gemeinsam mit seinen adjazenten Kanten in die vorhandene Zeichnung eingepasst werden muss. Zusätzlich werden noch theoretische Schranken für die Knickzahl und Fläche der Zeichnung untersucht.

In [BFG⁺97] wird das System „InteractiveGIOTTO“ vorgestellt. Das System beruht auf dem punktförmigen Zeichenmodell, womit wieder der Knotengrad auf vier beschränkt ist. In diesem System fügt der Benutzer am Bildschirm neue Knoten und Kanten zum Graphen hinzu. In einem Orthogonalisierungsschritt wird dann eine neue Zeichnung berechnet, in der dann auch alle neuen Kanten orthogonal sind. Es gibt keine Garantie, dass die resultierende Zeichnung planar ist, auch wenn diese Forderung relativ leicht zu erreichen wäre. Allerdings bleiben die Einbettung, die Kantenschnitte, die Kantenknicke und die Richtung, in der eine Kante einen Knoten verlässt, erhalten. Dem Erhalt der „mental map“ wird also ein großes Gewicht eingeräumt, da sich die Flächenformen nicht ändern. Die Beschränkung auf den Knotengrad vier ist für praktische Beispiele ein Problem und planare Zeichnungen sollten aus unserer Sicht wenn möglich auch planar gezeichnet werden.

Der aus unserer Sicht am weitesten fortgeschrittene Algorithmus zum interaktiven orthogonalen Zeichnen von planaren Graphen ist der Algorithmus aus [BGW00]. Dieser Algorithmus kann die Planarität sicherstellen, er kann Knoten mit beliebigem Grad verarbeiten und eine Änderung eines Kantenverlaufes wird, so weit möglich, vermieden. Andere Algorithmen erfüllen diese wichtigen Anforderungen entweder nicht oder schlechter.

Die Frage zur Laufzeit haben wir schon oben angesprochen. Außerdem können Knoten zwischen verschiedenen Zeichenschritten ihre Größe ändern oder Zeichnungen sind nur quasiorthogonal. Interaktive Anforderungen zur Erhaltung der „mental map“ fließen zwar in die Bewertungsfunktion mit ein, sie werden aber nie garantiert, so dass für einen Anwender keine verlässlichen Invarianten erkennbar sind.

Insbesondere der letzte Punkt und die Änderung der Knotengrößen sind uns so wichtig, dass wir diese Punkte bei unserem Verfahren ändern wollen. Knoten sollen ihre Größe nicht ändern (außer der Anwender verlangt das), es soll einfach verständliche Invarianten geben, die bei jedem Zeichenschritt erhalten bleiben. Das gilt auch dann, wenn dadurch statische Qualitätsanforderungen weniger gut erfüllt werden können.

Damit haben wir die wichtigsten Arbeiten im Umfeld unseres neuen Ansatzes zum interaktiven orthogonalen Zeichnen beschrieben. Wir werden uns nun im nächsten Kapitel überlegen, wie Flächen am besten geformt sein sollten, damit sie für interaktives, orthogonales Zeichnen geeignet sind. Danach gehen wir auf die Einfügeoperationen bei unserem Verfahren ein.

4 Theoretische Grundlagen

In diesem Kapitel definieren wir zunächst Anforderungen (Restriktionen) für einen interaktiven Zeichenalgorithmus, die sicherstellen sollen, dass die „mental map“ eines Benutzers nicht zerstört wird. Sie müssen so einfach sein, dass ein Anwender sie schnell und leicht versteht (erlernen kann) und intuitiv erkennt, welche Auswirkungen sich für aufeinander folgende Zeichnungen ergeben. Diese Restriktionen müssen auch flexibel genug sein, um eine qualitativ hochwertige Zeichnung zu ermöglichen. Wir geben außerdem das von uns primär untersuchte Qualitätskriterium mit an, mit dem *qualitativ hochwertige Zeichnungen* formal definiert werden können.

Restriktionen für einen interaktiven Zeichenalgorithmus sollen sicherstellen, dass sich zwei aufeinander folgende Zeichnungen möglichst wenig unterscheiden. Dieses Ziel kann auf zwei verschiedenen Wegen erreicht werden.

Beim ersten Lösungsansatz wird die bisherige Zeichnung des Graphen L_1 mit der Folgezeichnung L_2 verglichen, und eine Funktion $\delta(L_1, L_2)$ misst den Unterschied zwischen beiden Zeichnungen. Wenn $\delta(L_1, L_2)$ bestimmte, näher zu definierende Anforderungen erfüllt, dann erfüllt auch der Zeichenalgorithmus die Anforderungen.

Bei der zweiten Alternative werden für einen Graphen bestimmte Eigenschaften definiert, die jede Zeichnung erfüllen muss. Diese Einschränkungen für die Zeichnungen werden so definiert, dass je zwei Zeichnungen ähnlich sind, wenn sie die geforderten Eigenschaften erfüllen.

Außerdem sind für einen interaktiven Zeichenalgorithmus auch Änderungen an der Struktur des Graphen notwendig (z. B. Einfügen von Knoten und Kanten). Eine Funktion zur Änderung der Struktur des Graphen muss neben dem Graph dann noch eine Menge von Restriktionen in eine neue Menge überführen. Auch diese Transformation muss so gestaltet sein, dass eine Zeichnung des geänderten Graphen ähnlich zu Zeichnungen des bisherigen Graphen ist. Wieder ist es hier aber nicht notwendig, eine schon vorhandene Zeichnung zu berücksichtigen.

Wir haben für unseren Ansatz die zweite Alternative gewählt. Sie erlaubt es, eine Zeichnung eines Graphen zu bestimmen, ohne eine schon gegebene Zeichnung zu berücksichtigen. Es muss dann nur sichergestellt werden, dass jede Zeichnung die festgelegten Eigenschaften erfüllt.

Im Rest dieses Kapitels geben wir zunächst die Restriktionen für einen Zeichenalgorithmus an. Im zweiten Schritt beschäftigen wir uns mit der Frage, wie die Restriktionen abgespeichert werden können und wie sie von Einfügeoperationen verändert werden dürfen.

Im dritten Schritt definieren wir das betrachtete Qualitätskriterium. Wir untersuchen dann, wie gut ein beliebiger interaktiver Zeichenalgorithmus zum orthogonalen Zeichnen von planaren Graphen bzgl. dieses Qualitätskriterium unter Berücksichtigung der vorher definierten Restriktionen sein kann.

4.1 Restriktionen für Zeichnungen

Die zu wählenden Restriktionen müssen sicherstellen, dass die „mental map“ des Anwenders nicht zerstört wird. Dazu muss die folgende Anforderung erfüllt werden:

- Die Restriktionen bilden eine Einschränkung für die Menge der erlaubten Zeichnungen des Graphen. Die Einschränkungen sind dabei so zu wählen, dass sich je zwei Zeichnungen, die sie berücksichtigen, *ähnlich* sind und damit die „mental map“ des Anwenders nicht verletzen.

Wir betrachten die folgenden Einschränkungen für einen Zeichenalgorithmus. Sie werden in der nachfolgenden Definition noch formal gefasst.

1. Die Zeichnung entspricht einer gegebenen Einbettung.
2. Die relative Lage von je zwei adjazenten Knoten ist in x- und y-Richtung gegeben und die Zeichnung entspricht diesen Einschränkungen. Das heißt, falls zwei Knoten v_1 und v_2 durch eine Kante verbunden sind, dann ist definiert, ob v_1 in einer Zeichnung des Graphen links oder rechts und ober- oder unterhalb von v_2 liegen muss.

Warum halten wir diese beiden Anforderungen für wichtig und gut? Zunächst ist der Erhalt der Einbettung ein wichtiges Charakteristikum für interaktive Zeichenalgorithmen, das auch in [MELS95] gefordert wird. Ändert sich die Einbettung, dann ändert sich auch die Struktur der Zeichnung, was keinesfalls das Ziel bei einem interaktiven Zeichenalgorithmus sein kann.

Damit bleibt die Anforderung, dass die relative Lage adjazenter Knoten nicht geändert wird. Es ist eine Abschwächung der Forderung aus [MELS95], dass die relative Lage von Knoten nicht geändert werden darf. Wir begnügen uns deshalb mit der Abschwächung, weil die Form einer Fläche damit, abgesehen von Stauchungen und Dehnungen, erhalten bleibt. Das ist unserer Meinung nach ausreichend, denn Knoten sind in der Regel dann durch eine Kante verbunden, wenn sie in einer Beziehung zueinander stehen. Damit orientiert man sich in einem Graphen eher über Kanten, als zwischen zwei beliebige Knoten.

Warum ist diese Anforderung gut? Nehmen wir an, wir haben in einer Landkarte die Stadt Passau im Blick und suchen eine Stadt München. Wenn wir einen netten Menschen zur Seite haben, der uns helfen will, dann wird er uns sagen, dass München links unterhalb von Passau liegt. Oder wir erhalten die Information, dass Passau nach

links oben von einer Autobahn verlassen wird, die dann nach links unten abknickt. Im Verlauf der Autobahn liegt München.

Mit jeder dieser beiden Informationen ist es leicht, München zu finden. Deshalb orientiert sich die zweite Anforderung an dieser Idee der Suche in einer Landkarte. Ein Anwender kann genau auf die gleiche Art in einem Graphen navigieren. Die Lagebeziehungen zwischen den Knoten sind in der „mental map“ des Anwenders gespeichert. Auf Grund dieser Lageinformationen kann er durch die Zeichnung des Graphen navigieren, genau in der gleichen Art, wie er den Weg von Passau nach München findet.

Wir fassen die obigen Ideen zur Definition des *Restriktionensystem* eines Graphen zusammen.

Definition 1 (Restriktionensystem)

Das *Restriktionensystem* eines planaren Graphen besteht aus

1. einer Einbettung E des Graphen und
2. zwei Attributen \preceq_x und \preceq_y für jede Kante $e = \{v_1, v_2\}$.

Eine Zeichnung eines Graphen *respektiert* das Restriktionensystem des Graphen, wenn

1. die Zeichnung der Einbettung E entspricht und
2. für jede Kante $e = \{v_1, v_2\}$ in der Zeichnung für die Lagebeziehungen der Knoten v_1 und v_2 gilt: $v_1.x \preceq_x v_2.x$ und $v_1.y \preceq_y v_2.y$

Ein Restriktionensystem heißt *gültig*, wenn es eine planare orthogonale Zeichnung gibt, die das Restriktionensystem respektiert.

Im Folgenden schreiben wir kurz $G = (V, E, R)$ für einen Graphen G mit Knotenmenge V , Kantenmenge E und Restriktionensystem R . ■

Die Anordnung der adjazenten Knoten wird durch die Vergleichsoperation \preceq festgelegt. Im Rahmen dieser Arbeit werden wir hier die Vergleichsoperationen $<$ und \leq betrachten.

Wir verwenden die folgende Datenstruktur zur Speicherung eines Restriktionensystems:

1. Die Kanten um einen Knoten sind in einer zyklischen Liste abgelegt. Die Außenfläche ist bekannt und wird in einem separaten Attribut abgelegt.
2. Jede Kante $\{v_1, v_2\}$ trägt zwei boolsche Attribute \preceq_x und \preceq_y , die festlegen, ob v_1 oberhalb (\preceq_y) und links (\preceq_x) von v_2 liegt.¹ Falls $\preceq_x = true$, dann liegt v_1 links von v_2 , andernfalls rechts. Falls $\preceq_y = true$, dann liegt v_1 unterhalb von v_2 , andernfalls oberhalb.²

¹ Damit dieses Attribut auch mathematisch korrekt ist, müssten v_1 und v_2 als Repräsentation einer Kante eigentlich geordnet sein. Wir gehen davon aus, dass dies auf Grund der Repräsentation in zwei verschiedenen Attributen bei einer Implementierung gegeben ist. Andernfalls müsste festgelegt werden, welcher von beiden Knoten „der Erste“ ist.

² Wir nehmen an, dass der Ursprung des Koordinatensystems in der linken unteren Ecke liegt.

In Abbildung 4.1 sind auf der linken Seite die Lagebeziehungen eines Restriktionensystems eines Graphen gegeben. Die Darstellung der Einbettung wurde zur Vereinfachung weggelassen, sie kann aus einer der beiden Abbildungen 4.1(b) oder 4.1(c) abgelesen werden. Die booleschen Werte *true* und *false* werden wie üblich durch die Buchstaben *t* und *f* repräsentiert. Daneben sind zwei mögliche Zeichnungen des Graphen dargestellt, die das Restriktionensystem respektieren.

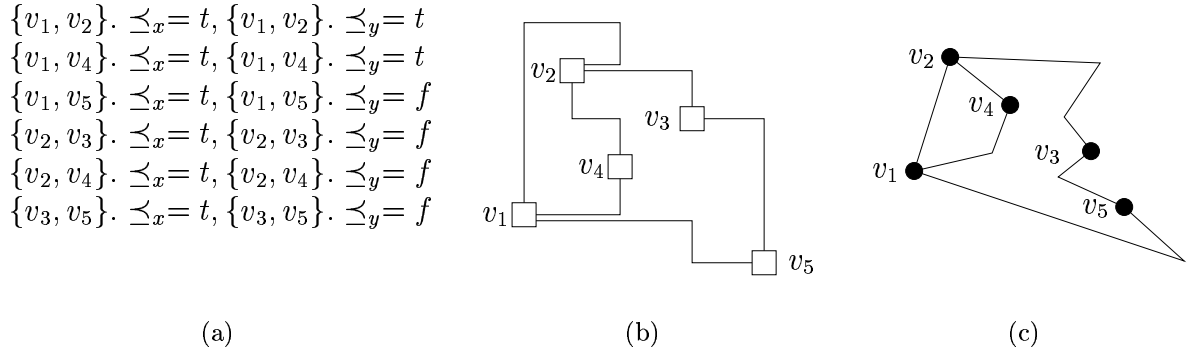


Abbildung 4.1: Lagebeziehungen eines Restriktionensystem eines Graphen und zwei Zeichnungen des Graphen, die das Restriktionensystem respektieren.

Beide Zeichnungen in Abbildung 4.1 respektieren das angegebene Restriktionensystem. Trotzdem unterschieden sie sich deutlich. Wir betrachten in dieser Arbeit jedoch ausschließlich orthogonale Zeichnungen. Wenn, neben der Fixierung auf orthogonale Zeichnungen, auch noch der Verlauf der Kanten eingeschränkt wird, dann erhalten wir sehr ähnliche Zeichnungen, wie in Abbildung 4.2 zu sehen ist. Beide Zeichnungen entsprechen wieder dem Restriktionensystem aus Abbildung 4.1. Allerdings handelt es sich um zwei orthogonale Zeichnungen und die Kantenverläufe (also die Abfolge von waage- und senkrechten Kantenstücken) sind ähnlich. Nur bei der Kante $\{v_1, v_5\}$ haben wir zwei verschiedene Kantenverläufe gewählt.

Neben dem Restriktionensystem werden wir also zusätzlich daran interessiert sein, die Kantenverläufe so wenig wie möglich zu ändern. Diese Anforderung werden wir dadurch berücksichtigen, dass es einen *bevorzugten* Kantenverlauf gibt. Nur wenn eine orthogonale planare Zeichnung mit diesem Kantenverlauf nicht möglich ist, dann wird ein anderer, jedoch auch fester Verlauf gewählt.

4.2 Änderungen an der Graphstruktur

Bisher haben wir die Einschränkungen für einen Zeichenalgorithmus betrachtet. Da wir uns aber mit interaktiven Zeichenalgorithmen beschäftigen, ist dies noch nicht ausreichend. Wir müssen definieren, wie ein gegebenes Restriktionensystem eines Graphen verändert wird, wenn die Struktur des Graphen verändert wird. Da wir uns primär mit

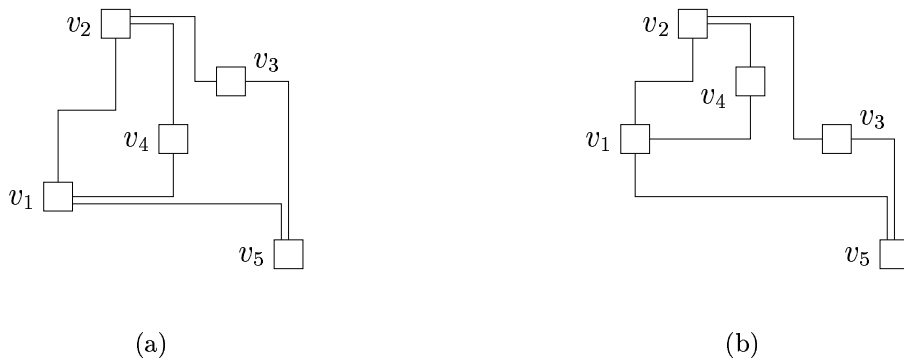


Abbildung 4.2: Vergleich von zwei orthogonalen Zeichnungen, die das gleiche Restriktionssystem respektieren bei weitgehender Fixierung des Kantenverlaufes.

den Einfügeoperationen *InsertEdge* und *InsertNode* beschäftigen, wie sie in Kapitel 2 beschrieben sind, werden wir die Einschränkungen bzgl. dieser Operationen definieren.

Unsere Anforderungen sind informell wie folgt:

- Wenn eine Kante eingefügt wird, dann muss die Einbettung des Graphen geändert werden. Dabei wird gefordert, dass die neue Einbettung ohne der zusätzlichen Kante der alten Einbettung entspricht.
- Wenn durch das Ersetzen einer Kante $\{v_1, v_2\}$ ein neuer Knoten v und zwei Kanten $\{v_1, v\}$ und $\{v, v_2\}$ entstehen, dann soll die Einbettung dergestalt erhalten bleiben, dass die neuen Objekte die alte Kante ersetzen. Das heißt, wenn die beiden neuen Kanten und der neue Knoten als eine Kante interpretiert werden, dann entspricht die neue Einbettung der alten.
- Beim Erzeugen eines neuen Knotens durch das Ersetzen einer Kante erhalten die beiden neuen Kanten die gleichen Lagebeziehungen wie die alte Kante. Das heißt, für v_1 und v_2 gelten nach der Funktion die gleichen Lagebeziehungen wie vorher.
- Sowohl beim Erzeugen eines neuen Knotens als auch beim Erzeugen einer neuen Kante bleiben die Restriktionen bzgl. der Lagebeziehungen von adjazenten Knoten erhalten. Das heißt, für keine Kante ändert sich das Attribut \preceq_x oder \preceq_y .

Die Anforderungen sind so definiert, dass sich ein Restriktionssystem eines Graphen möglichst wenig ändert. Lediglich die neuen Objekte werden in der Graphstruktur ergänzt, das Restriktionssystem wird nur ergänzt und nicht geändert. Das heißt, die Anforderungen zur Erhaltung der „mental map“ werden sehr gut erfüllt.

Die Änderungen müssen natürlich so gestaltet sein, dass das Restriktionssystem auch nach den Einfügeoperationen gültig ist, wenn der Graph mit dem Restriktionssystem vor der Einfügeoperation gültig war. Die Frage, ob das überhaupt möglich ist, wird im Rahmen dieser Arbeit beantwortet.

In der folgenden Definition fassen wir die Anforderungen für die Einfügeoperationen formal zusammen.

Definition 2 (Restriktionenerhaltende Einfügeoperation)

Gegeben sei ein Graph $G^{(1)} = (V^{(1)}, E^{(1)}, R^{(1)})$ mit einem gültigen Restriktionensystem $R^{(1)}$.

Eine Operation *InsertEdge*, deren Ergebnis ein Graph $G^{(2)} = (V^{(2)}, E^{(2)}, R^{(2)})$ ist und die eine neue Kante e erzeugt, heißt *restriktionenerhaltende Einfügeoperation*, wenn folgende Bedingungen erfüllt sind:

1. $\forall e \in E^{(1)}: e^{(1)}. \preceq_x = e^{(2)}. \preceq_x \wedge e^{(1)}. \preceq_y = e^{(2)}. \preceq_y$
2. Die Einbettungen von $G^{(1)} = (V^{(1)}, E^{(1)}, R^{(1)})$ und $G^{(2)} = (V^{(2)}, E^{(2)} \setminus \{e\}, R^{(2)})$ sind identisch.

Eine Operation *InsertNode*, deren Ergebnis ein Graph $G^{(2)} = (V^{(2)}, E^{(2)}, R^{(2)})$ ist und die aus einer Kante $e = \{v_1, v_2\}$ in $E^{(1)}$ den Knoten v und die Kanten e_1 und e_2 erzeugt, heißt *restriktionenerhaltende Einfügeoperation*, wenn sie folgende Eigenschaften erfüllt:

1. $\forall e \in E^{(1)} \setminus \{e\}: e^{(1)}. \preceq_x = e^{(2)}. \preceq_x \wedge e^{(1)}. \preceq_y = e^{(2)}. \preceq_y$
2. Die Einbettungen von $G^{(1)} = (V^{(1)}, E^{(1)}, R^{(1)})$ und $G^{(2)} = (V^{(2)}, E^{(2)}, R^{(2)})$ sind identisch, wenn in $G^{(2)}$ der Knoten v und die Kanten $e_1 = \{v_1, v\}$ und $e_2 = \{v, v_2\}$ durch eine einzelne Kante $e' = \{v_1, v_2\}$ ersetzt werden.
3. $e. \preceq_x = e_1. \preceq_x = e_2. \preceq_x \wedge e. \preceq_y = e_1. \preceq_y = e_2. \preceq_y$. ■

Damit haben wir alle Anforderungen definiert. Als Einfügeoperationen betrachten wir restriktionenerhaltende Einfügeoperationen. Ein Zeichenalgorithmus muss aus einem Graph mit einem gültigen Restriktionensystem eine Zeichnung im Kandinsky-Modell konstruieren, die das Restriktionensystem respektiert.

Wir werden nun noch das betrachtete Qualitätskriterium festlegen, um anschließend zu untersuchen, welche Resultate ein interaktives Verfahren bestenfalls erzielen kann, wenn die hier definierten Einschränkungen betrachtet werden.

4.3 Qualitätskriterien

Wir beschäftigen uns mit orthogonalen Zeichenalgorithmen. Durch die Konstruktion des Graphen mit den betrachteten Einfügeoperationen erhalten wir stets planare Graphen mit einer planaren Einbettung, für die auch eine planare Zeichnung erzeugt werden kann.

Für orthogonale Zeichnungen gilt neben der Planarität die Minimierung der Kantenknicke als wichtigstes Qualitätskriterium ([PCJ95]). Deshalb soll die Anzahl der Kantenknicke klein sein.

Bei einem interaktiven Zeichenalgorithmus gilt insbesondere, dass der Graph Knoten für Knoten und Kante für Kante aufgebaut wird. Der Benutzer hat also in der Regel

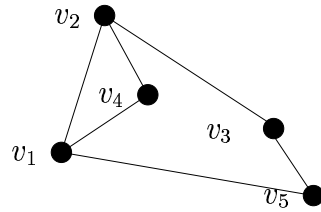


Abbildung 4.3: Graphische Darstellung eines Restriktionensystems.

immer eine eingeschränkte Sicht auf eine beschränkte Menge von Kanten, nämlich die neue(n) Kante(n), die beim Einfügen eines Knotens oder einer Kante entstehen. Deshalb ist es uns sehr wichtig, dass nicht nur die Gesamtzahl an Kantenknicken, sondern auch die maximale Anzahl an Knicken in einer Kante klein ist.

Damit erhalten wir die folgenden Qualitätskriterien für eine Zeichnung:

1. Die Zeichnung ist planar.
2. Die maximale Anzahl an Knicken je Kante ist minimal.
3. Die Gesamtanzahl an Kantenknicken ist minimal.

Planarität ist eine Minimalanforderung, die immer erfüllt sein muss. Damit bleibt die Minimierung der Anzahl an Kantenknicken als Optimierungskriterium.

Im letzten Abschnitt dieses Kapitels wollen wir uns mit der Frage beschäftigen, welche Resultate im Hinblick auf die Minimierung der Anzahl an Knicken in einer Kante bestenfalls erreicht werden können.

4.4 Theoretisch erreichbare Ergebnisse

Wir verwenden ab jetzt geradlinige Zeichnungen zur Darstellung eines Restriktionensystems. In den Zeichnungen liegt ein Knoten v_1 immer dann links (unterhalb) von einem benachbarten Knoten v_2 , wenn nach dem Restriktionensystem $v_1.x \preceq v_2.x$ ($v_1.y \preceq v_2.y$) gilt. Die Einbettung der Zeichnung entspricht ebenfalls der Einbettung des Restriktionensystems. Die Abbildung 4.3 zeigt als Beispiel die graphische Beschreibung des Restriktionensystems vom Graph in Abbildung 4.2(a).

Wir beschäftigen uns mit folgender Frage:

Gegeben sei ein Graph mit einem Restriktionensystem und eine Zahl k . Welche Anforderungen an das Restriktionensystem müssen erfüllt sein, damit man zwei beliebige Knoten in einer gemeinsamen Fläche durch eine Kante verbinden kann, so dass es eine planare Zeichnung im Kandinsky-Modell gibt, in der die neue Kante mit maximal k Knicken gezeichnet werden kann.

Die folgende Definition formalisiert diese Fragestellung.

Definition 3 (k -verträglich)

Ein Restriktionensystem eines Graphen G heißt k -verträglich, wenn für je zwei beliebige Knoten v_1 und v_2 einer Fläche f beim Einfügen einer neuen Kante zwischen v_1 und v_2 gilt:

1. e verläuft innerhalb der Fläche f und
2. es existiert eine orthogonale planare Zeichnung des Graphen im Kandinsky-Modell, die das Restriktionensystem respektiert und
3. in dieser Zeichnung wird e mit maximal k Knicken gezeichnet. ■

Im Folgenden wollen wir untersuchen, wie groß dieses k mindestens sein muss, wenn ein interaktiver Zeichenalgorithmus mit unseren Anforderungen konstruiert werden soll. Wir untersuchen also die Frage, wie groß die maximale Anzahl an Knicken in einer Kante aus theoretischen Überlegungen mindestens werden kann.

Da wir an einer unteren Schranke für die Anzahl an Kantenknicken interessiert sind, betrachten wir hier die Ordnungsrelation \leq . Bei einem Algorithmus, der durch $<$ stärker eingeschränkt wird, muss die untere Schranke dann ebenfalls gelten. Die folgenden Überlegungen gelten also sowohl für die Verwendung von $<$ als auch für die Verwendung von \leq an Stelle von \preceq .

Wir betrachten zunächst noch keine Sequenz von Operationen, wie es bei einem interaktiven Zeichenverfahren nötig wäre, sondern nur eine einzige Operation. Wir beschäftigen uns also mit der Frage, wie die Knoten einer Fläche relativ zueinander liegen müssen, damit die Fläche k -verträglich ist. Die Überlegungen werden aber auf eine Beschreibung von erlaubten Flächenformen hinauslaufen und eine Einfügeoperation muss von erlaubten Formen wieder in erlaubte abbilden. Damit folgt automatisch der Übergang zu einer Sequenz von Operationen.

Wir wissen, dass es statische Zeichenverfahren gibt, die für einen gegebenen planaren Graphen eine planare Zeichnung des Graphen im Kandinsky-Modell erzeugen, in der jede Kante maximal einen Knick hat ([FKK96, Bie97]). Auf Grund dessen untersuchen wir zunächst, welche Voraussetzungen ein Restriktionensystem eines Graphen erfüllen muss, damit eine beliebige neue orthogonale Kante zwischen zwei Knoten in einer Fläche f mit maximal einem Knick eingefügt werden kann, wenn die neue Kante in f verlaufen muss. Wir suchen also Eigenschaften des Restriktionensystems, die für 1-Verträglichkeit notwendig sind.

Wir betrachten zunächst für eine Fläche f jeweils nur je drei zyklisch aufeinander folgende Knoten. Welche durch das Restriktionensystem festgelegten räumlichen Anordnungen der Knoten verhindern, dass eine neue Kante zwischen den beiden äußeren der drei Knoten mit maximal einem Knicken gezeichnet werden kann? Abbildung 4.4 zeigt eine solche Anordnung. Wenn v_1 und v_2 in einer Zeichnung, die das Restriktionensystem respektiert, mit einer orthogonalen Kante in der Fläche f verbunden werden sollen, dann muss die Kante um den Knoten v herum geführt werden. Es sind also mindestens zwei Knicke notwendig und eine solche Anordnung muss verboten sein. Andere verbotene An-

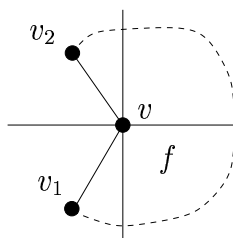


Abbildung 4.4: Verbotene Anordnungen von je drei aufeinander folgenden Knoten in der Fläche f , wenn eine Kante zwischen v_1 und v_2 mit maximal einem Knick gezeichnet werden soll.

ordnungen entstehen durch Spiegelungen dieser Situation. Die folgende Definition legt ein Modell fest, das auf dieser Überlegung basiert. Es schränkt die erlaubte räumliche Anordnung der Knoten um jede einzelne Fläche ein.

Definition 4 (4-Sektorenmodell)

Gegeben sei ein planarer, zweifach zusammenhängender Graph G mit einem Restriktionssystem R . Dann ist G mit R im *4-Sektorenmodell*, wenn für alle Flächen f außer der Außenfläche von G gilt:

1. Sei (v_1, \dots, v_n) die Sequenz der Knoten um f im Uhrzeigersinn.³ Dann existiert eine Aufteilung von (v_1, \dots, v_n) in vier aufeinander folgende Teilsequenzen S_1, \dots, S_4 für die folgende Bedingungen erfüllt sind:
 - In S_1 liegt der jeweilige Nachfolgeknoten rechts oberhalb:
 $\forall v_i \in S_1 : v_i.x \preceq v_{i+1}.x \wedge v_i.y \preceq v_{i+1}.y,$
 - In S_2 liegt der jeweilige Nachfolgeknoten rechts unterhalb:
 $\forall v_i \in S_2 : v_i.x \preceq v_{i+1}.x \wedge v_i.y \succeq v_{i+1}.y,$
 - In S_3 liegt der jeweilige Nachfolgeknoten links unterhalb:
 $\forall v_i \in S_3 : v_i.x \succeq v_{i+1}.x \wedge v_i.y \succeq v_{i+1}.y,$
 - In S_4 liegt der jeweilige Nachfolgeknoten links oberhalb:
 $\forall v_i \in S_4 : v_i.x \succeq v_{i+1}.x \wedge v_i.y \preceq v_{i+1}.y.$
2. Es existieren mindestens zwei Sequenzen i und $i + 2$ (mit $i \in \{1, 2\}$) die nicht leer sind.

Im Folgenden bezeichnen wir die Knoten in S_1, \dots, S_4 mit *Sektor 1, ..., Sektor 4* der Fläche f . Eine Kante e liegt in Sektor i ($i \in \{1, \dots, 4\}$) der Fläche f , falls der in der zyklischen Ordnung um f erste Endknoten von e in Sektor i liegt.

³ o. B. d. A. sei v_1 der Knoten, der in $\{v_1, \dots, v_n\}$ links liegt. Das heißt, er hat in der Menge keinen Nachbarn, der wegen der Lagebeziehungen links von ihm liegen muss.

Anstelle von G und das zugehörige Restriktionensystem R liegen im 4-Sektorenmodell sagen wir im Folgenden auch kurz, der Graph G und das Restriktionensystem R sind *legal*. ■

Bemerkung 1

Im Rahmen der Arbeit benötigen wir für einen gegebenen Sektor i oft den Sektor $i + 1$ oder Sektor $i + 2$. Falls $i + 1$ oder $i + 2$ größer als vier ist, dann ist jeweils der Sektor bzgl. Modulo-Rechnung gemeint. Falls also z. B. $i = 4$ gilt, dann ist Sektor $i + 1$ Sektor 1 und Sektor $i + 2$ ist Sektor 2. □

In der Definition wird gefordert, dass es immer ein Paar i und $i + 2$ von Sektoren gibt, das nicht leer ist. Dies ist notwendig, da andernfalls keine Zeichnung existiert, die das Restriktionensystem respektiert. Nehmen wir zum Beispiel an, dass in einer Fläche f die beiden Sektoren 3 und 4 leer sind und Sektor 1 und 2 nicht. Sei v_1 der erste Knoten in Sektor 1 und v_2 der letzte Knoten in Sektor 2. Wegen des zweifachen Zusammenhanges des Graphen muss es in f eine weitere Kante oder Sequenz von Kanten zwischen v_1 und v_2 geben. Da aber v_1 links von v_2 liegt, muss v_2 in Sektor 3 oder 4 liegen und wir erhalten einen Widerspruch. Es kann also nicht sein, dass zwei aufeinander folgende Sektoren leer sind.

Bild 4.5 zeigt zwei Flächen, bei denen die Anordnung der Knoten dem 4-Sektorenmodell entspricht. In der ersten sind alle Sektoren belegt, in der zweiten ist der Sektor 2 der Fläche leer. Die Knoten eines Sektors sind jeweils markiert. Die einem Knoten im Uhrzeigersinn folgende Kante liegt im gleichen Sektor wie der Knoten.

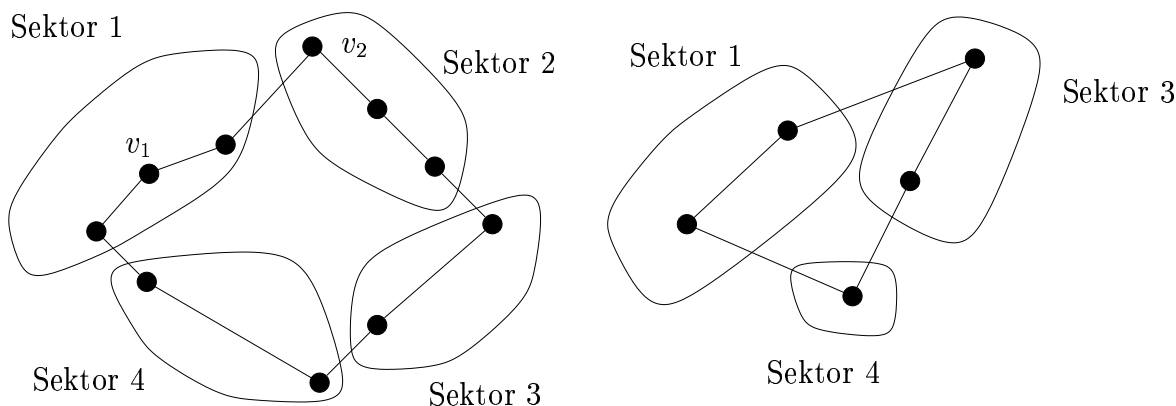


Abbildung 4.5: Zwei Beispiele für das 4-Sektorenmodell.

In diesen Bildern fällt auf, dass die graphische Darstellung eines Restriktionensystems, das dem 4-Sektorenmodell entspricht, stark an konvexe Zeichnungen erinnert. Allerdings ist der Begriff der Konvexität strenger als das 4-Sektorenmodell. Das heißt, jede konvexe Zeichnung kann als Darstellung eines Restriktionensystems, das dem 4-Sektorenmodell entspricht, verstanden werden.

Denn es gilt: Wenn eine Zeichnung konvex ist, dann gibt es in keiner Fläche einen stumpfen Innenwinkel. Wenn diese Zeichnung als Darstellung eines Restriktionensystems

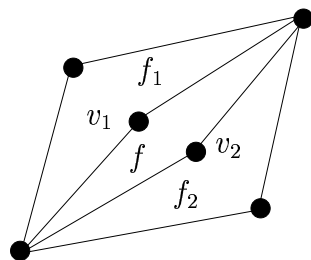


Abbildung 4.6: Konvexität ist restriktiver als das 4-Sektorenmodell.

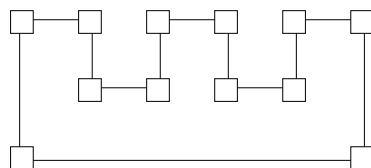


Abbildung 4.7: Orthogonale Repräsentation, die zwar „turn regular“ ist, aber die Anordnung der Knoten entspricht nicht dem 4-Sektorenmodell.

verstanden wird, dann führt diese Eigenschaft genau dazu, dass die Anforderungen des 4-Sektorenmodells erfüllt sind, denn Anordnungen wie in Abbildung 4.4 sind nicht möglich.

Allerdings gibt es Restriktionensysteme, die dem 4-Sektorenmodell entsprechen, für die es aber keine Darstellung gibt, die konvex ist (vgl. Abbildung 4.6). Dieses Restriktionensystem mit der durch das Restriktionensystem gewählten Anordnung der Knoten kann nicht konvex dargestellt werden, denn die innere Fläche f kann keine leere Fläche haben. Damit ist der Winkel an entweder dem Knoten v_1 oder v_2 an der Innenseite von f spitz und als Folge davon muss es in einer der beiden Flächen f_1 oder f_2 einen stumpfen Winkel geben.

Die Eigenschaften des 4-Sektorenmodells sind auch zu den Anforderungen bei „turn regularity“ nicht identisch. Wenn in einer Fläche alle 4 Sektoren belegt sind, dann sind die Knoten in gegenüberliegenden Sektoren weder in x- noch in y-Richtung geordnet. Andererseits gibt es Graphen, die zwar „turn regular“ sind, aber nicht dem 4-Sektorenmodell entsprechen (Abbildung 4.7).

Im Rahmen der Arbeit benötigen wir oft die Information, in welche Richtung eine Kante einen Knoten auf Grund des Restriktionensystem verlassen muss. Dazu dient die folgende Definition.

Definition 5 ($d(e, v)$)

Gegeben sei ein Graph G und ein Restriktionensystem R für den Graphen. Sei $e = \{v, v'\}$ eine zu v adjazente Kante. Die Funktion $d(e, v)$ liefert die Richtung, in der die Kante e den Knoten v im Restriktionensystem R verlässt. Damit gilt:

1. Falls $e. \preceq_x = t \wedge e. \preceq_y = t$, dann gilt $d(e, v) = \nearrow$
2. Falls $e. \preceq_x = t \wedge e. \preceq_y = f$, dann gilt $d(e, v) = \searrow$

3. Falls $e. \preceq_x = f \wedge e. \preceq_y = t$, dann gilt $d(e, v) = \nearrow$

4. Falls $e. \preceq_x = f \wedge e. \preceq_y = f$, dann gilt $d(e, v) = \swarrow$ ■

Mit Hilfe der Definition des 4-Sektorenmodells kann leicht überprüft werden, wann ein gegebener Graph mit seinem Restriktionensystem dem 4-Sektorenmodell entspricht. Hierfür müssen die durch die Kanten induzierten Lagebeziehungen untersucht werden. Das Restriktionensystem ist legal, wenn in jeder Fläche eine Partition der Knoten in die vier Sektoren gefunden werden kann und die Einbettung planar ist. Das folgende Lemma zeigt das.

Lemma 1

Gegeben sei ein planarer, zweifach zusammenhängender Graph G mit einem Restriktionensystem R . Es kann in $O(n)$ berechnet werden, ob der Graph und das Restriktionensystem dem 4-Sektorenmodell entsprechen.

BEWEIS Für einen Graph kann in $O(n)$ berechnet werden, ob die gegebene zyklische Ordnung der Kanten um einen Knoten einer planaren Einbettung entspricht.

Damit muss noch festgestellt werden, ob die Lagebeziehungen der adjazenten Knoten dem 4-Sektorenmodell entsprechen.

Dazu laufen wir zweimal um jede Fläche f , außer der Außenfläche. Dies kann in Summe in $O(n)$ durchgeführt werden, wenn bei jeder Kante die jeweilige Vorgänger- und Nachfolgerkante für die beiden Endknoten bekannt ist.

Beim ersten Durchlauf wird ein Knoten gesucht, der bzgl. der Fläche f keine von links einlaufende Kante hat. Für diesen Knoten v mit den beiden in f zu ihm adjazenten Kanten $e_1 = \{v, v_1\}$ und $e_2 = \{v, v_2\}$ muss also gelten, dass $e_1. \preceq_x = t$ und $e_2. \preceq_x = t$. Wenn es keinen solchen Knoten gibt, dann ist das Restriktionensystem nicht legal. Wenn es einen solchen Knoten v gibt, dann muss dieser Knoten der erste in Sektor 1 oder 2 sein. Damit kann der zweite Lauf um die Fläche bei v beginnen und es wird überprüft, ob die Zuordnung der aufeinander folgenden Knoten dem 4-Sektorenmodell entspricht.

Außerdem kann nach dem Umlauf in $O(1)$ festgestellt werden, ob mindestens zwei gegenüberliegende Sektoren nicht leer sind, da die belegten Sektoren beim Umlauf gespeichert werden können. ■

Damit haben wir das 4-Sektorenmodell definiert und gezeigt, wann ein zu einem Graph gehörendes Restriktionensystem diesem Modell genügt.

Im Weiteren werden wir nun untersuchen, inwieweit das eben definierte Zeichenmodell für interaktive, orthogonale Zeichenalgorithmen geeignet ist. Dazu zeigen wir zunächst, dass das 4-Sektorenmodell für 1-verträgliche Zeichnungen notwendig ist. Wenn es also einen interaktiven Zeichenalgorithmus geben sollte, bei dem nur restriktionenerhaltende Einfügeoperationen erlaubt sind und jede Kante maximal einen Knick haben soll, dann müssen die generierten Flächenformen dem 4-Sektorenmodell entsprechen.

Satz 2 (1-verträglich \Rightarrow 4-Sektorenmodell)

Gegeben sei ein Graph G mit einem Restriktionensystem. Wenn das Restriktionensystem eines Graphen 1-verträglich ist, dann entspricht es dem 4-Sektorenmodell. Das 4-Sektorenmodell ist also notwendig für 1-verträgliche Zeichnungen.

BEWEIS Gegeben sei ein Graph G mit einem 1-verträglichen Restriktionensystem.

Nehmen wir an, dass dieses Restriktionensystem nicht dem 4-Sektorenmodell entspricht. Dann muss es eine Fläche f geben, in der der Sektor von mindestens einem Knoten v nicht der Abfolge des 4-Sektorenmodells entspricht, wir erhalten also eine Situation wie in Abbildung 4.4 (also z.B. der Vorgänger liegt in Sektor 1 und der Nachfolger von v liegt aber links oberhalb, so dass v eigentlich in Sektor 4 liegen müsste).

Sei v_1 der Vorgänger von v in f und v_2 der Nachfolger von v in f . Dann können v_1 und v_2 nicht 1-verträglich sein, denn jede orthogonale Kante zwischen den beiden Knoten innerhalb der betrachteten Fläche hätte mindestens zwei Knicke. ■

Auf Grund von Satz 2 wissen wir, dass 1-verträgliche Zeichnungen von Graphen das 4-Sektorenmodell implizieren. Nun stellt sich die Frage, ob jedes Restriktionensystem, das dem 4-Sektorenmodell entspricht, auch 1-verträglich ist. Wir untersuchen also, ob das 4-Sektorenmodell hinreichend für 1-verträgliche Zeichnungen ist.

Lemma 2 (4-Sektorenmodell $\not\Rightarrow$ 1-verträglich)

Es gibt Restriktionensysteme im 4-Sektorenmodell, die nicht 1-verträglich sind.

BEWEIS Gegeben sei ein Graph G mit dem Restriktionensystem, dessen graphische Repräsentation in Abbildung 4.8 dargestellt ist.

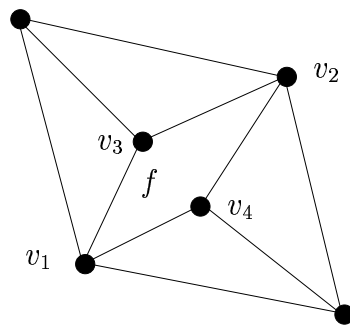


Abbildung 4.8: Restriktionensystem eines Graphen, das im 4-Sektorenmodell liegt, aber v_1 und v_2 sind nicht 1-verträglich.

Dieses Restriktionensystem entspricht dem 4-Sektorenmodell, denn jede Fläche entspricht dem 4-Sektorenmodell. Es gibt aber keine Zeichnung im Kandinsky-Modell, die das Restriktionensystem respektiert und in der die beiden Knoten v_1 und v_2 durch eine orthogonale Kante mit einem Knick in der Fläche f verbunden werden können:

In Abbildung 4.9 sind die beiden möglichen orthogonalen Kantenverläufe zwischen zwei Knoten v_1 und v_2 eingezeichnet. Wegen der Lagebeziehungen des Restriktionensystems des Graphen müssen aber sowohl v_3 als auch v_4 innerhalb des Rechteckes liegen, das durch diese beiden Kantenverläufe definiert wird. ■

Wir haben gezeigt, dass das 4-Sektorenmodell zwar notwendig, aber nicht hinreichend für 1-verträgliche Zeichnungen ist. Damit stellt sich die Frage, ob dies ein Mangel des

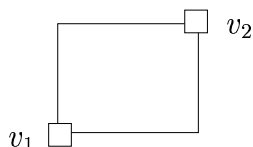


Abbildung 4.9: Mögliche Kantenverläufe mit einem Knick zwischen zwei Knoten.

Modells oder ein generelles Problem bei interaktiven orthogonalen Algorithmen ist. Die Antwort gibt der folgende Satz, in dem gezeigt wird, dass es für bestimmte Graphen unabhängig vom aktuellen Restriktionensystem eine Sequenz von Einfügeoperationen gibt, so dass bei den von uns verwendeten Anforderungen für die Einfügeoperationen ein nicht 1-verträgliches Restriktionensystem entsteht.

Satz 3 (Algorithmus mit maximal einem Knick je Kante ist nicht möglich)

Für die Zeichnung eines Graphen seien die Anforderungen des Kandinsky-Modells mit festen Knotengrößen gegeben. Einfügeoperationen müssen restriktionenerhaltend sein. Es existieren Graphen, bei denen — unabhängig vom anfänglich gewählten Restriktionensystem — nach einer Sequenz von Einfügeoperationen mindestens ein Knotenpaar, das in einer gemeinsamen Fläche liegt, nicht mehr 1-verträglich ist.

BEWEIS Wir führen den Beweis durch Angabe eines Beispiels. Dazu sei der Graph aus Abbildung 4.10⁴ gegeben.

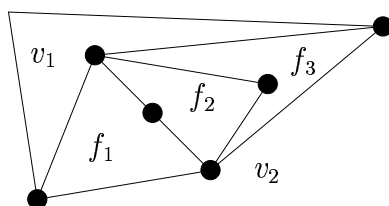


Abbildung 4.10: Graph, bei dem nach zwei Einfügeoperationen die Knoten v_1 und v_2 nicht mehr 1-verträglich sind.

Wir betrachten die Knoten v_1 und v_2 . Diese beiden Knoten können nur in den Flächen f_1, f_2 und f_3 durch eine Kante verbunden werden, da dies die einzigen gemeinsamen Flächen sind. Außerdem gibt es, unabhängig von der gewählten Zeichnung, maximal zwei orthogonale Kanten mit einem Knick, die v_1 und v_2 verbinden, nämlich⁵ zuerst ein waage- und dann ein senkrechtes Segment (l_1), oder umgekehrt (l_2). Diese beiden Verläufe sind schon in Abbildung 4.9 dargestellt.

Wir betrachten die möglichen Kantenverläufe l_1 und l_2 . Die relative Lage adjazenter Knoten darf bei Einfügeoperationen nicht geändert werden.

⁴ Hier handelt es sich *nicht* um die Darstellung eines Restriktionensystems.

⁵ beginnend am linken Knoten

Für die Knoten, die sowohl mit v_1 als auch mit v_2 verbunden sind, gilt: Auf Grund der Lagebeziehungen des Restriktionensystems liegen sie entweder innerhalb oder außerhalb des durch l_1 und l_2 definierten Rechteckes. Sie können aber auch bei Einfügeoperationen nie von innerhalb nach außerhalb wechseln und umgekehrt, wenn die Einfügeoperationen restriktionenerhaltend sind.

Außerdem gilt: In einer Zeichnung des obigen Graphen, die das Restriktionensystem respektiert, können die Knoten v_1 und v_2 nur dann in einer der Flächen f_1, \dots, f_3 mit einer Kante mit maximal einem Knick verbunden werden, wenn einer der beiden übrigen Knoten, die die Fläche begrenzen, innerhalb des Rechteckes und einer außerhalb liegt.

Von den drei Flächen können nur zwei diese Anforderung erfüllen. Betrachte dazu die in einer Zeichnung mittlere der drei Flächen. Wenn die mittlere die Anforderung erfüllt, dann müssen bei einer der beiden äußeren Flächen beide Knoten außerhalb des Rechteckes liegen (Abbildung 4.11). Und wenn die beiden äußeren Flächen die Anforderung erfüllen, dann liegen bei der inneren Fläche die beiden Knoten außer v_1 und v_2 innerhalb des Rechteckes.

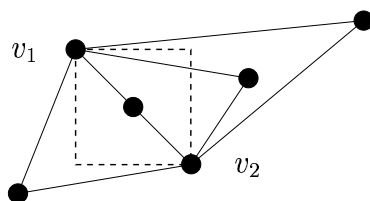


Abbildung 4.11: Mögliche Kanten mit einem Knick zwischen v_1 und v_2 . Die übrigen Knoten können nicht über die gestrichelten Linien wandern.

Nun werden in den beiden Flächen, die die Anforderung erfüllen, die beiden von v_1 und v_2 verschiedenen Knoten durch eine Kante verbunden. Es verbleibt eine gemeinsame Fläche für v_1 und v_2 . In dieser Fläche können aber in keiner Zeichnung, die das Restriktionensystem respektiert, die beiden Knoten v_1 und v_2 durch eine Kante mit maximal einem Knick verbunden werden. Damit ist der Graph mit seinem Restriktionensystem nach zwei Einfügeoperationen nicht mehr 1-kompatibel, unabhängig vom Initial gewählten Restriktionensystem. ■

Aus Sicht der Knickzahl ist damit das 4-Sektorenmodell für interaktives, orthogonales Zeichnen von zweifach zusammenhängenden Graphen gut geeignet, da zwar Kanten mit zwei Knicken nötig sind, was aber generell nicht vermeidbar ist. Wir werden bei der Generierung einer orthogonalen Zeichnung in Kapitel 6 sehen, dass maximal $|F|$ Kanten zwei Knicke besitzen. Alle anderen Kanten haben einen Knick.

Es stellt sich noch die Frage, ob alle planaren zweifach zusammenhängenden Graphen in diesem Modell darstellbar sind. In den nächsten Kapiteln geben wir einen interaktiven Zeichenalgorithmus an, mit dem für jeden zweifach zusammenhängenden, planaren Graphen eine Zeichnung im 4-Sektorenmodell erzeugt werden kann. Damit gilt, dass diese Graphen in dem Modell darstellbar sind.

Wir haben also gezeigt, dass das 4-Sektorenmodell nicht zwingend 1-verträglich ist. Andererseits haben wir aber auch gezeigt, dass es kein interaktives Zeichenverfahren geben kann, bei dem jede Kante maximal einen Knick hat, wenn gefordert wird, dass die relative Lage von je zwei adjazenten Knoten nicht geändert werden darf.

Wie wir im weiteren Verlauf sehen werden, genügt uns das 4-Sektorenmodell auch mit einer weiteren Einschränkung⁶: Von nun an fordern wir in Zeichnungen von Graphen zusätzlich, dass eine Kante auf der rechten Seite des linken Knotens beginnt. Wir verschärfen damit die Invarianten des Zeichenalgorithmus. Wir tun das, da sich unserer Meinung nach orthogonale Zeichnungen wesentlich ähnlicher sind, wenn die Kantenverläufe nicht beliebig variiert werden. Alle obigen Überlegungen bleiben gültig, da sie nur auf relativen Knotenpositionen und nicht auf Kantenverläufen basieren.

Im folgenden Kapitel werden wir die restriktionenerhaltenden Einfügeoperationen für einen interaktiven Zeichenalgorithmus entwickeln, die vom 4-Sektorenmodell wieder in das 4-Sektorenmodell abbilden.

⁶ d.h., die zusätzliche Einschränkung produziert in der O -Notation keine zusätzlichen Kantenknicke.

5 Einfügeoperationen

In diesem Abschnitt wird die Implementierung der Operationen

1. $Init()$: Erzeugen eines initialen Graphen.
2. $InsertEdge(v_1, v_2, f)$: Einfügen einer Kante zwischen v_1 und v_2 in der Fläche f .
3. $InsertNode(e)$: Einfügen eines neuen Knotens auf Kante e .

beschrieben.

Mit Hilfe der hier definierten Funktionen werden der Graph und das Restriktionensystem Schritt für Schritt aufgebaut. Sie stellen also die Komponente dar, die einen interaktiven Zeichenalgorithmus entstehen lässt. Die Operationen bilden einen Graphen mit einem Restriktionensystem, das die Anforderungen des 4-Sektorenmodells erfüllt, wieder in einen Graphen mit einem Restriktionensystem im 4-Sektorenmodell ab. Dabei müssen die Operationen die Einschränkungen erfüllen, die wir für restriktionerhaltende Einfügeoperationen definiert haben.

Wir werden nach einführenden Definitionen zunächst zeigen, dass Einfügeoperationen existieren, die unseren Anforderungen genügen. Im zweiten Schritt beschäftigen wir uns mit der Frage, mit welcher Komplexität diese Funktionen implementiert werden können.

In den folgenden Überlegungen wird oft danach unterschieden, ob ein Knoten am Anfang oder an einer beliebigen anderen Stelle eines Sektors liegt. Aus diesem Grund definieren wir hier die Begriffe Sektorstartknoten und Sektormittelknoten.

Definition 6 (Sektorstartknoten)

Ein Knoten v heißt *Sektorstartknoten* i in Fläche $f \Leftrightarrow v$ ist in der zyklischen Ordnung der Knoten um Fläche f der erste Knoten in Sektor i . ■

Definition 7 (Sektormittelknoten)

Ein Knoten v heißt *Sektormittelknoten* in der Fläche f , wenn er in f kein Sektorstartknoten ist. ■

5.1 Einfügeoperationen

Wir beschreiben zuerst die Initialisierungsfunktion und dann die Funktionen $insertNode$ und $insertEdge$.

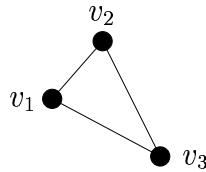


Abbildung 5.1: Graph, der durch die Initialisierungsfunktion erzeugt wird.

5.1.1 Erzeugen eines initialen Graphen

In Abbildung 5.1 ist eine graphische Repräsentation des initialen Graphen mit seinem Restriktionensystem dargestellt.

Der Graph und das Restriktionensystem erfüllen die Anforderungen des 4-Sektorenmodells, da der Graph nur eine Innenfläche hat und diese Fläche dem 4-Sektorenmodell entspricht. (v_1 liegt in Sektor 1, v_2 in Sektor 2 und v_3 in Sektor 4.)

5.1.2 Einfügen eines Knotens

Wir definieren zunächst die Funktion und zeigen dann, dass die Funktion restriktionenerhaltend ist und dass ein legales Restriktionensystem wieder in ein legales abgebildet wird.

Definition 8 (*insertNode*)

Gegeben sei ein Graph $G^{(1)} = (V^{(1)}, E^{(1)}, R^{(1)})$ mit einem Restriktionensystem, das dem 4-Sektorenmodell entspricht. Die Funktion $insertNode(e)$ erzeugt wie folgt den Graph $G^{(2)} = (V^{(2)}, E^{(2)}, R^{(2)})$:

Die Kante $e = \{v_1, v_2\}$ wird aus dem Graph entfernt. Es werden ein neuer Knoten v und die beiden neuen Kanten $e_1 = \{v_1, v\}$ und $e_2 = \{v, v_2\}$ in den Graph eingefügt. Die beiden neuen Kanten und der neue Knoten ersetzen die Kante e in der Einbettung des Graphen. Es gilt $e_1 \cdot \preceq_x = e_2 \cdot \preceq_x = e \cdot \preceq_x$ und $e_1 \cdot \preceq_y = e_2 \cdot \preceq_y = e \cdot \preceq_y$ ■

Das folgende Lemma zeigt, dass die eben definierte Funktion restriktionenerhaltend ist.

Lemma 3 (*insertNode* ist restriktionenerhaltend und erhält das 4-Sektorenmodell)

Die in Definition 8 definierte Funktion ist restriktionenerhaltend. Sie führt einen Graph mit einem legalen Restriktionensystem wieder in einen Graph mit einem legalen Restriktionensystem über.

BEWEIS Die Funktion ist restriktionenerhaltend, denn die Festlegung der Attribute entspricht den Anforderungen aus Definition 2.

Unter der Voraussetzung, dass das Restriktionensystem vor der Ausführung der Funktion dem 4-Sektorenmodell entspricht, entspricht auch der neue Graph mit dem neuen Restriktionensystem dem 4-Sektorenmodell:

Es gibt nur zwei Flächen, in denen Änderungen stattfinden, nämlich die beiden Flächen, in denen die Kante e lag. Wir müssen die geforderte Eigenschaft also nur für diese

beiden Flächen zeigen. Dazu muss die Partition der Knoten in die 4 Sektoren gesucht werden. Sei dazu v' derjenige zu e adjazente Knoten, der in der zyklischen Ordnung der Knoten um die Fläche der erste der beiden Endknoten der Kante e ist. Der neue Knoten v liegt auf Grund der Definition der Attribute \preceq_x und \preceq_y der beiden neu erzeugten Kanten im gleichen Sektor wie v' , so dass nur dieser Sektor verlängert wird. Alle anderen Sektoren bleiben unverändert. Damit entspricht das erzeugte Restriktionssystem dem 4-Sektorenmodell. ■

Abbildung 5.2 illustriert die Funktion *InsertNode*. Die Änderung der Halbordnungen wird in Abbildung 5.2(a) dargestellt. Dabei symbolisiert die gepunktete Linie die Abhängigkeiten in x-Richtung und die gestrichelte die Abhängigkeiten in y-Richtung. Abbildung 5.2(b) zeigt die neue Einbettung. Die Pfeile symbolisieren hier die zyklische Ordnung der Kanten um die Endknoten.

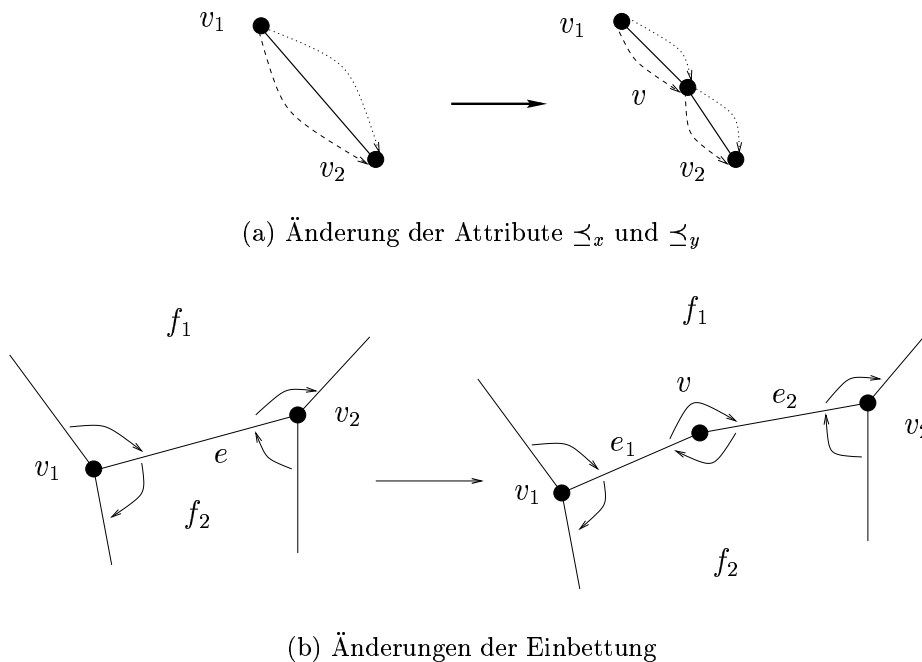


Abbildung 5.2: Änderungen am Restriktionssystem beim Einfügen eines Knotens.

5.1.3 Einfügen einer Kante

Auch hier definieren wir zunächst die Funktion und zeigen dann, dass sie restriktionenerhaltend ist und dass der erzeugte Graph mit seinem Restriktionssystem dem 4-Sektorenmodell entspricht.

Definition 9 (*insertEdge*)

Gegeben sei ein Graph $G^{(1)} = (V^{(1)}, E^{(1)}, R^{(1)})$ mit einem Restriktionensystem, das dem 4-Sektorenmodell entspricht. Die Funktion $insertEdge(v_1, v_2, f)$ (wobei v_1 und v_2 in f liegen müssen) erzeugt wie folgt den Graph $G^{(2)} = (V^{(2)}, E^{(2)}, R^{(2)})$:

Es wird eine neue Kante $\{v_1, v_2\}$ erzeugt, die die bisherige Fläche f in zwei neue Flächen f_1 und f_2 teilt.

Bzgl. der Attribute \preceq_x und \preceq_y unterscheiden wir drei verschiedene Fälle:

1. Die Knoten v_1 und v_2 sind wegen der Transitivität von \preceq auf Grund der Kanten, die um die Fläche f angeordnet sind, sowohl in x- als auch in y-Richtung geordnet:
Die Belegung der Attribute \preceq_x und \preceq_y wird aus dieser Ordnung übernommen.

2. Die Knoten v_1 und v_2 sind wegen der Transitivität von \preceq auf Grund der Kanten, die um die Fläche f angeordnet sind, entweder in x- oder in y-Richtung geordnet:
In der Richtung, in der die Knoten v_1 und v_2 geordnet sind, muss die Kante e die gleiche Ordnung induzieren. Die Lageabhängigkeit in der anderen Richtung kann beliebig gewählt werden.

3. Die Knoten v_1 und v_2 sind wegen der Transitivität von \preceq auf Grund der Kanten, die um die Fläche f angeordnet sind, weder in x- noch in y-Richtung geordnet.¹

Wenn v_1 in Sektor 1 und v_2 in Sektor 3 liegt, dann soll $\{v_1, v_2\}. \preceq_x = t$ und $\{v_1, v_2\}. \preceq_y = f$ gelten. Wenn v_1 in Sektor 2 und v_2 in Sektor 4 liegt, dann soll $\{v_1, v_2\}. \preceq_x = f$ und $\{v_1, v_2\}. \preceq_y = f$ gelten. ■

Die Abbildungen 5.3, 5.4 und 5.5 zeigen für jeden Fall der Definition ein Beispiel.

In Abbildung 5.3 liegen die Knoten v_1 und v_2 in der Fläche f im gleichen Sektor. Die Kanten in dem Sektor, die zwischen v_1 und v_2 liegen, haben alle die gleiche Belegung der Attribute \preceq_x und \preceq_y . Die Relation \preceq ist transitiv und deshalb sind v_1 und v_2 sowohl in x- als auch in y-Richtung geordnet. Abbildung 5.4 zeigt ein Beispiel, in dem die beiden Knoten bei alleiniger Betrachtung der Kanten um die Fläche f nur in x-Richtung geordnet sind. Die Knoten v_1 und v_2 liegen hier in Sektor 1 und Sektor 2. Da aber für die Kanten in Sektor 1 $\preceq_y = t$ und in Sektor 2 $\preceq_y = f$ gilt, sind die beiden Knoten in y-Richtung nicht geordnet. Abbildung 5.5 zeigt ein Beispiel, in dem die Knoten weder in x- noch in y-Richtung geordnet sind.

Es ist noch zu zeigen, dass die eben beschriebene Funktion restriktionenerhaltend ist und einen Graph mit einem legalen Restriktionensystem wieder in einen Graph mit einem legalen Restriktionensystem abbildet.

Lemma 4 (*insertEdge* ist restriktionenerhaltend und erhält das 4-Sektorenmodell)

Die in Definition 9 definierte Funktion ist restriktionenerhaltend. Sie führt einen Graph mit einem legalen Restriktionensystem wieder in einen Graph mit einem legalen Restriktionensystem über.

¹ Der Fall ist nur dann möglich, wenn alle Sektoren belegt sind und wenn die beiden Knoten v_1 und v_2 in gegenüberliegenden Sektoren liegen. Die Knoten können auch keine Sektorstartknoten sein.

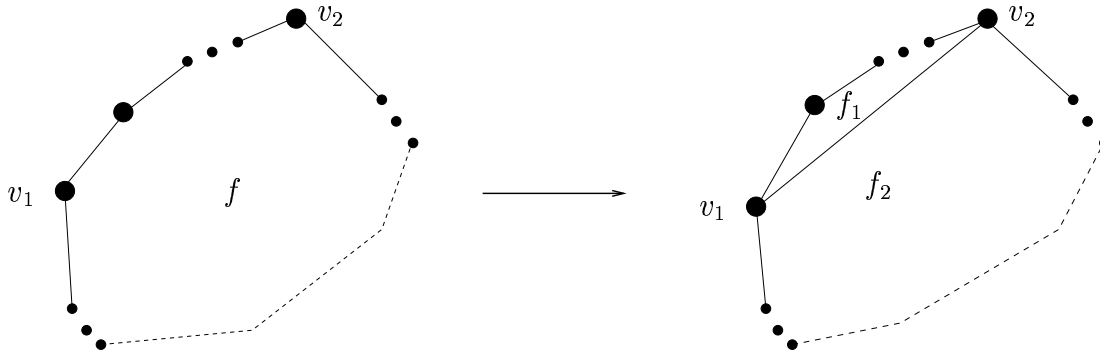


Abbildung 5.3: Einfügen einer neuen Kante zwischen zwei Knoten, die in x- und y-Richtung geordnet sind.

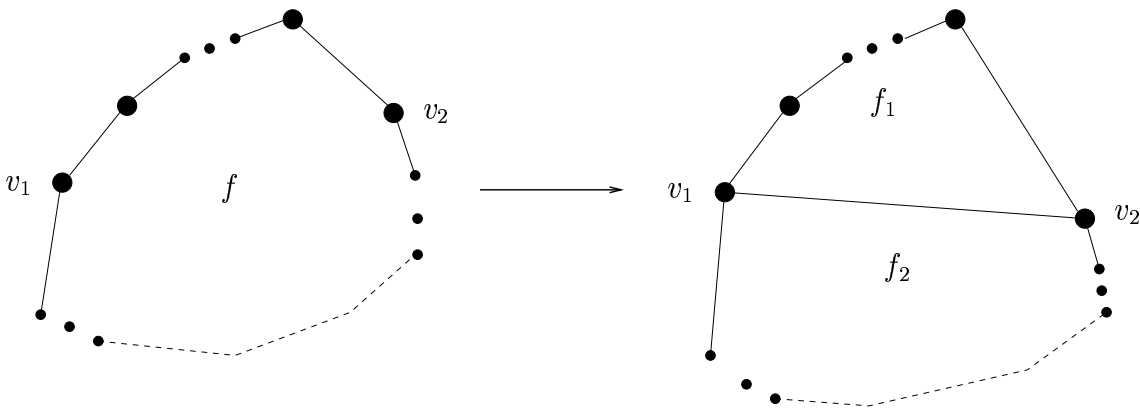


Abbildung 5.4: Einfügen einer neuen Kante zwischen zwei Knoten, die nur in x- oder y-Richtung geordnet sind.

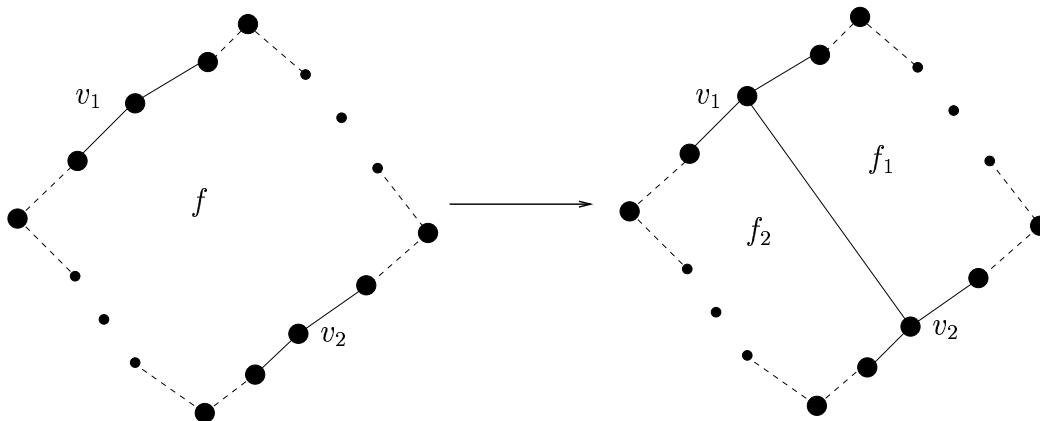


Abbildung 5.5: Einfügen einer neuen Kante zwischen zwei Knoten, die weder in x- noch in y-Richtung geordnet sind.

BEWEIS Die Funktion ist restriktionenerhaltend:

Die Attribute \preceq_x und \preceq_y werden für keine der Kanten geändert und die vorhandene Einbettung wird nur um die neue Kante ergänzt.

Weiter ist zu zeigen, dass unter der Voraussetzung, dass das Restriktionensystem vor der Ausführung der Funktion dem 4-Sektorenmodell entsprochen hat, auch der neue Graph mit dem neuen Restriktionensystem dem 4-Sektorenmodell entspricht:

Da wir außer in der Fläche f keine Änderungen vornehmen, müssen die gewünschten Eigenschaften nur für die beiden neuen Flächen f_1 und f_2 gezeigt werden. Sei immer f_1 die Fläche, die von der Sequenz der Knoten von v_1 bis v_2 und f_2 die Fläche, die von der Sequenz der Knoten von v_2 bis v_1 — jeweils in zyklischer Ordnung der Knoten um die Fläche f — gebildet wird. Wir unterscheiden die drei Fälle aus der Definition.

1. Die Knoten v_1 und v_2 sind wegen der Transitivität von \preceq auf Grund der Kanten, die um die Fläche f angeordnet sind, sowohl in x- als auch in y-Richtung geordnet.

Damit liegen die beiden Knoten im gleichen Sektor i , bzw. einer der beiden Knoten ist Sektorstartknoten des Sektors, der Sektor i folgt. Sei o.B.d.A. v_1 der erste der beiden Knoten, wenn f in zyklischer Ordnung, beginnend bei Sektorstartknoten i , umlaufen wird.

In der Fläche f_2 existiert die gleiche Menge an Sektoren wie sie schon in f existiert hat. Denn die neue Kante e ersetzt in f_2 die Sequenz der Knoten und Kanten zwischen v_1 und v_2 aus f . Auch die Anordnung der Knoten entspricht dem 4-Sektorenmodell, denn es wird nur die Sequenz der Kanten durch eine neue Kante mit der selben Richtung ersetzt.

In der Fläche f_1 existieren zwei Sektoren i und $i + 2$ und damit entspricht diese Fläche ebenfalls dem 4-Sektorenmodell. Denn die Sequenz der Kanten zwischen v_1 und v_2 liegt in Sektor i . Und die Kante von v_2 nach v_1 hat die dazu inversen Relationen. Damit liegen die Kante und v_2 in Sektor $i + 2$.

2. Die Knoten v_1 und v_2 sind wegen der Transitivität von \preceq auf Grund der Kanten, die um die Fläche f angeordnet sind, entweder in x- oder in y-Richtung geordnet.

Da v_1 und v_2 durch die Kanten um f nur entweder in x- oder y-Richtung geordnet sind, müssen die Knoten der Sequenz von v_1 bis v_2 und von v_2 bis v_1 jeweils mindestens in zwei verschiedenen Sektoren liegen. Also sind in jeder der beiden neuen Flächen mindestens zwei Sektoren belegt.

Wir zeigen, dass die Knoten um die beiden neuen Flächen in die 4 Sektoren partitioniert werden können.

Für beide Flächen f_1 und f_2 gilt, dass die Anordnung der Knoten durch die Kanten, die schon vor der Einfügeoperation existierten, dem 4-Sektorenmodell entspricht. Die neue Kante e ersetzt Teile von Sektoren. Wenn die Kante e in einem der Sektoren liegt, die ersetzt werden, dann entspricht die Anordnung der Knoten immer noch dem 4-Sektorenmodell, denn die bisherige Partition in die Sektoren bleibt gültig.

Betrachte die Fläche f_1 , die von der Sequenz s der Knoten von v_1 bis v_2 und der neuen Kante e gebildet wird. Wir unterscheiden zwei Fälle.

Fall 1: Seien v_1 und v_2 durch die Sequenz s und die Kanten zwischen diesen Knoten entweder in x- oder y-Richtung geordnet. Diese Ordnung wird für $e = \{v_1, v_2\}$ übernommen. Damit ist aber die Ordnung von v_2 zu v_1 genau invers und die neue Kante e muss in einem anderen Sektor liegen als die Knoten zwischen v_1 und v_2 . Da diese Knoten schon in zwei verschiedenen Sektoren liegen, sind insgesamt drei Sektoren nicht leer und es gibt ein nicht leeres Paar von Sektoren i und $i + 2$.

Fall 2: Seien v_1 und v_2 durch die Sequenz s und die Kanten zwischen diesen Knoten weder in x- noch in y-Richtung geordnet. Damit können v_1 und v_2 nicht in zwei aufeinander folgenden Sektoren liegen und müssen in zwei gegenüberliegenden Sektoren liegen. Es gibt also ein Paar von gegenüberliegenden Sektoren, das nicht leer ist.

Mit einer entsprechenden Argumentation erhält man, dass f_2 dem 4-Sektorenmodell entspricht.

3. Die Knoten v_1 und v_2 sind wegen der Transitivität von \preceq auf Grund der Kanten, die um die Fläche f angeordnet sind, weder in x- noch in y-Richtung geordnet:

Da dann v_1 und v_2 in gegenüberliegenden Sektoren liegen und jeweils keine Sektorstartknoten sein können, sind in beiden neuen Flächen zwei gegenüberliegende Sektoren nicht leer.

Wenn v_1 in f in Sektor 1 liegt und v_2 in Sektor 3, dann liegt e in f_1 in Sektor 4 und in f_2 in Sektor 2. Damit entspricht die Anordnung der Knoten dem 4-Sektorenmodell.

Eine entsprechende Argumentation gilt, wenn v_1 in Sektor 2 und v_2 in Sektor 4 liegt. ■

In den beiden nächsten Abschnitten werden zwei Implementierungen der Funktionen *InsertNode* und *InsertEdge* entwickelt, die sich bzgl. der Komplexität unterscheiden. Die erste Realisierung hat im schlimmsten Fall lineare Laufzeit, die zweite eine logarithmische.

5.2 Lineare Laufzeit

Wir beschreiben zunächst die verwendeten Datenstrukturen. Danach zeigen wir, wie mit ihrer Hilfe die Einfügeoperationen in linearer Laufzeit implementiert werden können.

5.2.1 Verwendete Datenstrukturen

Bei der Analyse der Laufzeit der Operationen *InsertNode* und *InsertEdge* gehen wir in diesem Abschnitt von der folgenden internen Repräsentation des Restriktionensystems aus:

1. Halbordnungen in x- und y-Richtung:

Knotenpaare sind nur dann in x- und y-Richtung geordnet², wenn sie durch eine Kante verbunden sind. Eine Kante speichert in einem Tupel $(node_1, node_2)$ seine beiden Endknoten. Damit genügt es, wenn bei jeder Kante zwei Attribute \preceq_x und \preceq_y gespeichert werden, für die gilt:

a) $\preceq_x = true \Leftrightarrow$ Es gilt $node_1 \preceq_x node_2$.

b) $\preceq_y = true \Leftrightarrow$ Es gilt $node_1 \preceq_y node_2$.

2. Einbettungen:

Jeder Knoten hat einen Verweis *edge* auf eine beliebige Kante, die zu ihm inzident ist. Zusätzlich werden die Kanten um jeden Knoten zyklisch doppelt verkettet. Die Kantenattribute $succ(node_1)$, $pred(node_1)$, $succ(node_2)$ und $pred(node_2)$ speichern dazu jeweils die Vorgänger- oder Nachfolger-Kante bzgl. des Knotens $node_1$ bzw. $node_2$ im Uhrzeigersinn.

Die gleiche Notation wird auch dann verwendet, wenn für einen Knoten v eine inzidente Kante e bekannt ist und der Vorgänger oder Nachfolger von e am Knoten v gesucht wird. So liefert also $e.succ(v)$ den zyklischen Nachfolger von e am Endknoten v .

3. Flächen:

Jede Fläche besitzt einen Zeiger *edge* auf eine beliebige Kante in der Fläche. Von dieser Kante (und nur von dieser) verweisen wir auch zurück auf die Fläche. Da jede Kante in zwei Flächen liegt, müssen für diese je Kante zwei Zeiger bereitgestellt werden. Wir verwenden dazu die Attribute f_1 und f_2 . Diese Zeiger werden in den meisten Fällen nicht benötigt, da wir nur von einer Kante auf die Fläche verweisen. Die Zeiger werden mit einem Wert \perp belegt, falls der Zeiger nicht benötigt wird.

Außerdem wird in den Flächenattributen im Attribut *node* derjenige Endknoten von Kante *edge* gespeichert, der in der zyklischen Ordnung der Knoten und Kanten um f nach der Kante *edge* folgt.

Um auf das Attribut a eines Objektes³ o zuzugreifen, verwenden wir die Schreibweise $o.a$. Abbildung 5.6 zeigt einen Ausschnitt aus einem Graphen. Für ausgewählte Objekte werden die Attribute durch Pfeile repräsentiert.

Mit Hilfe der obigen Datenstrukturen ist es möglich, eine Fläche im Uhrzeigersinn⁴ zu umlaufen. Wir verwenden die Funktion *opp*, um bei einer Kante e den zweiten Endknoten zu erhalten, wenn der erste bekannt ist. Der Wert kann aus den Attributen $(node_1, node_2)$ berechnet werden.

² wenn man von Abhängigkeiten auf Grund der Transitivität von \preceq absieht

³ Knoten, Kante oder Fläche

⁴ Auch im Gegenuhrzeigersinn, aber das benötigen wir hier nicht.

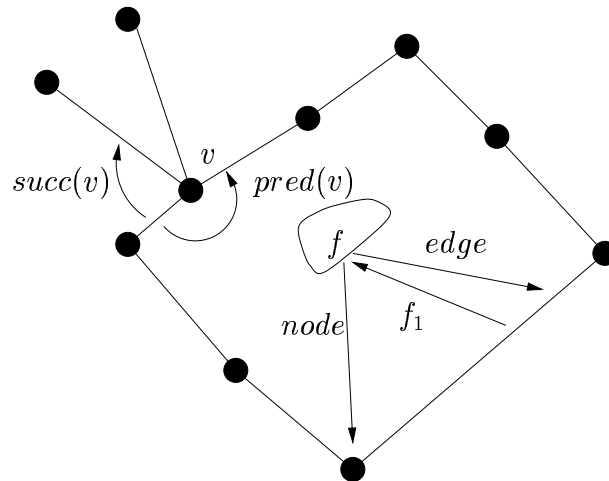


Abbildung 5.6: Die Attribute zur Verwaltung des Restriktionensystems.

Der von diesen Datenstrukturen benötigte Speicherplatz ist linear in der Menge der Knoten und Kanten, da jedes Objekt eine konstante Anzahl an Attributen besitzt. Wir beschreiben im Folgenden wie, und mit welcher Komplexität, die beiden Einfügeoperationen unter Zuhilfenahme der eben beschriebenen Repräsentation implementiert werden können.

5.2.2 Einfügen eines Knotens

Die Funktion zum Einfügen eines Knotens, wie sie in Abschnitt 5.1 definiert wurde, lautet $v = \text{InsertNode}(e)$. Unter der Voraussetzung, dass $e = \{v_1, v_2\}$ gilt, erhält man als Ergebnis der Einfügeoperation den neuen Knoten v und die beiden neuen Kanten $e_1 = \{v_1, v\}$ und $e_2 = \{v, v_2\}$.

Wir zeigen zunächst, wie die Attribute neu berechnet werden. Danach geben wir die Komplexität der Neuberechnung an.

1. Halbordnungen in x- und y-Richtung: Die neuen Kanten „erben“ die Lageabhängigkeiten der alten Kante e und erhalten damit also die identische Belegung dieser Attribute.
2. Einbettungen: Von der Kante e gibt es je einen Verweis auf die Vorgänger- und Nachfolgerkante bei beiden Endknoten $node_1$ und $node_2$. Damit kann zur Neuberechnung der Einbettung bei diesen Kanten der Verweis auf e durch e_1 bzw. e_2 ersetzt werden.

Bei dem neu erzeugten Knoten v ist der Vorgänger und Nachfolger von jeder der beiden neu erzeugten Kanten jeweils die andere.

3. Flächen: Falls bei Kante e einer der Verweise f_1 und f_2 auf eine Fläche ungleich \perp ist, dann muss für diese Fläche der Verweis auf die Kante e ersetzt werden, denn von

dieser Fläche wurde auf e verwiesen. Dazu wählen wir diejenige der neuen Kanten, die zu $f.node$ adjazent ist, den diese muss in der zyklischen Ordnung der Knoten und Kanten um f vor dem damit bezeichneten Knoten liegen. Zusätzlich muss von dieser Kante eines der beiden Attribute f_1 oder f_2 auf die Fläche verweisen.

Die Komplexität der Anpassung der Datenstrukturen ergibt sich aus folgender Überlegung: Alle Attribute können in konstanter Zeit neu berechnet werden, da die Objekte jeweils direkt vorliegen. Die Komplexität ist also $O(1)$. Damit ist auch die durchschnittliche Laufzeit der Operation konstant.

5.2.3 Einfügen einer Kante

Zum Einfügen einer Kante verwenden wir, wie in Abschnitt 5.1 schon beschrieben, die Funktion $e = InsertEdge(v_1, v_2, f)$, die zwischen den Knoten v_1 und v_2 in der Fläche f die neue Kante e einfügt. Dabei wird f in die beiden neuen Flächen f_1 und f_2 gespalten. Wir beschreiben im Folgenden die Anpassung der Datenstrukturen.

1. Halbordnungen: In Definition 9 hängen die Werte der Attribute \preceq_x und \preceq_y der neuen Kante e davon ab, ob die beiden Knoten durch die Menge der Kanten, die um die Fläche f angeordnet sind, in keiner, einer oder beiden Richtungen geordnet sind.

Zur Berechnung dieser Eigenschaft wird einmal im Uhrzeigersinn um die Fläche f gelaufen und die beiden Sequenzen von Kanten zwischen v_1 und v_2 und zwischen v_2 und v_1 untersucht. Wenn alle Kanten in einer Sequenz die gleiche Belegung eines der Attribute \preceq_x oder \preceq_y haben, dann sind die beiden Knoten auf Grund der Transitivität von \preceq geordnet.

Damit können die Attribute \preceq_x und \preceq_y der neuen Kante e gemäß Definition 9 festgelegt werden.

2. Einbettungen: Bei der Berechnung der Attribute \preceq_x und \preceq_y von der neuen Kante e wird die Fläche f einmal umlaufen.

Bei diesem Umlauf können die Kanten gespeichert werden, die sich an die Knoten v_1 und v_2 in Fläche f anschließen. Zwischen diesen Kanten wird die neue Kante jeweils eingefügt.

3. Flächen: Für die beiden neuen Flächen f_1 und f_2 müssen die Attribute $node$ und $edge$ festgelegt werden. Sei dazu f_1 die Fläche, in der v_1 zyklischer Nachfolger von v_2 ist. f_2 ist dann die Fläche, in der v_1 zyklischer Vorgänger von v_2 ist. Wir legen die Flächenattribute wie folgt fest:

$$\begin{aligned} f_1.edge &= e; f_1.node = v_1; e.f_1 = f_1 \\ f_2.edge &= e; f_2.node = v_2; e.f_2 = f_2 \end{aligned}$$

Die Komplexität der Einfügeoperationen ergibt sich durch die Komplexität der Berechnung der Kanten e_1 und e_2 , die den Knoten v_1 und v_2 folgen. Dazu müssen wir

schlimmstenfalls um eine ganze Fläche herumlaufen. Da in jeder Fläche $O(n)$ Knoten und Kanten liegen können, erhält man die worst case Komplexität $O(n)$.

Mit sehr einfachen Mitteln erhält man also eine lineare Laufzeit. Allerdings ist es aus theoretischer Sicht interessant, ob diese Laufzeit verbessert werden kann. Im nächsten Abschnitt entwickeln wir eine Datenstruktur, die auf einfachen balancierten Bäumen basiert und trotzdem $O(\log n)$ Laufzeit ermöglicht.

5.3 Erweiterungen für logarithmische Laufzeit

Wie wir im letzten Abschnitt gesehen haben, ergibt sich die lineare Laufzeit aus der Notwendigkeit, für einen Knoten die beiden Kanten zu finden, die an ihn in einer gegebenen Fläche anschließen, wenn eine neue Kante eingefügt wird. Alle anderen Operationen sind für die Laufzeit unkritisch, weil in konstanter Zeit berechenbar.

Unsere hier zu entwickelnde Datenstruktur muss also das Ziel haben, diese Information in maximal logarithmischer Zeit zur Verfügung zu stellen. Wir erreichen das, in dem wir die Information, welche Kanten an einen Knoten in der Fläche f anschließen, an zwei verschiedenen Orten speichern:

1. Jede Fläche speichert für ihre Sektorstartknoten, welche Kanten an sie anschließen. Da es in jeder Fläche nur maximal vier Sektorstartknoten gibt, ist dies auch für Update-Operationen beim Einfügen von Knoten und Kanten unkritisch.
2. Wenn ein Knoten in einer Fläche kein Sektorstartknoten ist, dann wird direkt in den Knotenattributen gespeichert, welche Kanten an den Knoten in dieser Fläche anschließen.

Wir stellen also fest: Wenn ein Knoten in einer Fläche kein Sektorstartknoten ist, dann werden die Kanten, die an den Knoten in der Fläche anschließen, direkt in den Knotenattributen gespeichert. Wenn hier $O(n)$ Kanten gespeichert werden müssen, weil der Knoten in n Flächen liegen kann, dann ist nichts gewonnen! Hier hilft uns die folgende Überlegung, die zeigt, dass ein Knoten immer nur in maximal zwei Flächen ein Sektormittelknoten sein kann und damit nur konstant viele Verweise zusätzlich nötig sind.

Lemma 5 (Jeder Knoten ist maximal zweimal Sektormittelknoten)

Es existiert kein Knoten v , der in mehr als zwei Flächen Sektormittelknoten ist.

BEWEIS Abbildung 5.7 zeigt eine Illustration zum Beweis. O. B. d. A. liege der Knoten v im Ursprung eines Koordinatensystems. v ist Sektorstartknoten in der Fläche f , wenn seine beiden Nachbarn in der Fläche f relativ zu ihm im gleichen oder in benachbarten Quadranten liegen. Also ist v Sektormittelknoten bzgl. f , wenn zwischen seinen beiden Nachbarn in f ein leerer Quadrant liegt. Soll nun ein Knoten in mehr als zwei Flächen Sektormittelknoten sein, dann müssten mindestens drei Quadranten (bedingt durch die mindestens drei Flächen, in denen v Sektormittelknoten ist) leer sein und zwischen diesen Quadranten müsste jeweils mindestens eine Kante liegen. Das ist nicht möglich, da nur vier Quadranten existieren. ■

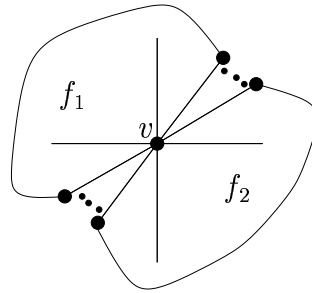


Abbildung 5.7: Knoten v kann in maximal zwei Flächen (f_1 und f_2) Sektormittelknoten sein.

Damit wäre unser Problem, welche Kanten an einen Knoten v in Fläche f anschließen, eigentlich schon gelöst. Falls der Knoten Sektorstartknoten in f ist, kann man die Information in konstanter Zeit aus den Flächenattributen abrufen. Sonst erhalten wir die Kanten aus den Knotenattributen, ebenfalls in konstanter Zeit.

Allerdings ergibt sich ein neues Problem. Da die neuen Attribute bei den Sektormittelknoten von den Flächen abhängig sind, ergibt sich beim Einfügen einer neuen Kante das Problem, dass die Fläche f durch zwei neue Flächen f_1 und f_2 ersetzt wird. Damit müssten die Attribute bei jedem Sektormittelknoten verändert werden, was nur in $O(n)$ möglich ist. Man beachte dabei, dass sich — außer bei v_1 und v_2 — die Kanten, die an den Knoten in der Fläche anschließen, nicht ändern, sondern nur die Verknüpfung mit einer Fläche. Hier ist also nur der Verweis auf die Fläche f durch einen Verweis auf f_1 oder f_2 zu ersetzen.

Zur Lösung dieses Problems wird die Verknüpfung aus Fläche und welche Kanten schließen in der Fläche an den Knoten an, nicht direkt in den Knotenattributen gespeichert. Statt dessen wird pro Fläche jeder der vier Sektoren jeweils durch einen balancierten Baum repräsentiert. Jede Fläche hat Verweise auf die ihr zugeordneten balancierten Bäume. Zusätzlich wird bei der Wurzel des Baumes ein Verweis auf die Fläche gespeichert. Jeder Sektormittelknoten eines Sektors hat einen Verweis auf ein ihm zugeordnetes Blatt des Baumes. Da jeder Knoten nur in zwei Flächen Sektormittelknoten sein kann, werden nur zwei Zeiger benötigt. Mit einem Durchlauf des Baumes von einem Blatt zu der Wurzel kann dann in logarithmischer Zeit die Fläche festgestellt werden, in der die Kante an den Knoten anschließt.

Wenn für einen Knoten v die Kanten gesucht werden, die an den Knoten in Fläche f anschließen, dann wird zuerst überprüft, ob der Knoten Sektorstartknoten in f ist. Wenn ja, dann erhalten wir die Informationen aus den Flächenattributen. Andernfalls laufen wir schlimmstenfalls zwei balancierte Bäume von einem Blatt bis zur Wurzel. Dort wird die dem Baum zugeordnete Fläche verglichen. Insgesamt ergibt sich also schlimmstenfalls logarithmischer Aufwand, wenn die balancierten Bäume maximal linear viele Objekte speichern.

Wir speichern auch für jede Kante einer Fläche einen Verweis in die Sektorbäume. Den Vorteil sehen wir bei der Beschreibung der Implementierung der Einfügeoperationen.

Damit existiert auch für jede Kante eines Sektors ein Blatt im Baum.

Zusätzlich muss der balancierte Baum so angereichert werden, dass die Reihenfolge von zwei gegebenen Knoten in einem Sektor einer Fläche bestimmt werden kann. Dies wird dann benötigt, wenn eine Kante zwischen zwei Knoten im gleichen Sektor eingefügt werden soll. Von der Reihenfolge hängen die Lagebeziehungen zwischen den Knoten ab. In den bisher beschriebenen Datenstrukturen sind die Lagebeziehungen aber nur für adjazente Knoten direkt gespeichert. Eine Realisierung des balancierten Baumes muss insgesamt folgende Funktionen zur Verfügung stellen:

Definition 10 (Sektorbaum)

Ein *Sektorbaum* ist eine Datenstruktur mit den Operationen

1. *init* ($()$): Erzeugen des leeren Sektorbaumes.
2. *insert* (o, o'): Einfügen des Elementes⁵ o nach o' . Es wird der Zeiger auf das Blatt zurückgegeben, das für Objekt o gültig ist.

Wenn o am Anfang des Sektorbaumes oder in den leeren Sektorbaum eingefügt werden soll, dann verwenden wir die Funktion ohne zweiten Parameter.

Die Funktion wird zum Einfügen von Knoten und Kanten benötigt, da ein Sektor dabei länger werden kann.

3. *split* (o): Teilen des Sektorbaumes in zwei Teile. Der neue Sektorbaum⁶ beginnt mit dem Element o , der ursprüngliche endet mit dem Element, das Vorgänger von o war. Element o liegt also nach der Operation nur im neu erzeugten Sektorbaum.

Die Funktion wird beim Einfügen einer Kante benötigt, da dabei ein Sektor in zwei Teile getrennt werden kann, die dann in verschiedenen Flächen liegen.

4. *delete* (o): Löschen von o aus dem Sektorbaum. Die Funktion wird ebenfalls beim Einfügen einer Kante benötigt, um neu entstandene Sektorstartknoten aus einem Sektor zu löschen.

5. *delete* (o_1, o_2): Ausschneiden des Teiles von o_1 bis o_2 . Dieser Teil wird als Ergebnis der Funktion zurückgegeben. Der neue Sektorbaum beginnt mit dem Element o_1 und endet mit dem Element o_2 . Die Elemente o_1 und o_2 und der Teil zwischen den Objekten sind nach der Operation nicht mehr Element des ursprünglichen Sektorbaumes.

Auch diese Funktion wird beim Einfügen einer Kante benötigt, wenn die beiden Endknoten in einem gemeinsamen Sektor liegen. Die neue Kante schneidet dann eine Sequenz aus der „Mitte“ eines Sektors.

6. *setInfo* (inf): Zuweisen einer Information inf an den Sektorbaum. Bei uns sind das die Fläche, zu der der Sektorbaum gehört, und die Nummer des Sektors.

⁵ In unserem Fall ist das ein Knoten oder eine Kante.

⁶ Er wird als Ergebnis der Funktion zurückgegeben.

7. *getInfo ()*: Holen der Information.
8. *before (o₁, o₂)*: Liefert *true*, falls *o₁* vor *o₂* liegt. ■

Ein Sektorbaum enthält in unserer Anwendung alle Knoten und alle Kanten, die in einem gegebenen Sektor einer Fläche liegen. Nur der Sektorstartknoten wird nicht in einem Sektorbaum repräsentiert. Damit beginnt und endet ein Sektorbaum jeweils mit einer Kante.

Im Folgenden zeigen wir eine mögliche Realisierung für einen Sektorbaum, der alle von uns geforderten Eigenschaften hat.

5.3.1 Sektorbäume

Wir beschreiben eine Realisierung von Sektorbäumen auf Basis von balancierten 2-3 Bäumen.

Satz 4 (Sektorbäume mit Funktionen in logarithmischer Laufzeit)

*Es existiert eine Implementierung von Sektorbäumen, in der die Funktionen *init()*, *insert(o, o')*, *split(o)*, *delete(o)*, *delete(o, o')*, *setInfo(info)*, *getInfo()* und *before(o, o')* logarithmische Laufzeit haben.*

BEWEIS In [AHU83] werden 2-3 Bäume erklärt, die die Funktionen *insert*, *delete*, *split* und *remove* in logarithmischer Zeit zur Verfügung stellen. Damit können die Operationen *insert(o₁, o₂)*, *delete(o)*, *delete(o₁, o₂)*, *split(o)* und *init()* in $O(\log k)$ implementiert werden. Da in den Sektorbäumen Knoten und Kanten eines planaren Graphen gespeichert sind, ist hier k ein konstantes Vielfaches der Knotenzahl des Graphen.

Damit fehlt die Beschreibung der Operationen *setInfo*, *getInfo* und *before(o₁, o₂)*.

Die Tiefe eines balancierten Baumes ist stets logarithmisch in der Anzahl der Blätter. Wie bei der abstrakten Beschreibung der Sektorbäume erklärt, erhält man für jedes eingefügte Objekt einen Zeiger auf das Blatt in dem Baum, das dem Knoten oder der Kante eindeutig zugeordnet ist.

In unserer speziellen Implementierung hat jeder Knoten nicht nur die Zeiger auf die Söhne, sondern auch auf seinen Vater. Diese zusätzlichen Zeiger können ebenso wie die Zeiger vom Vater auf den Sohn bei allen anderen Operationen mit aktualisiert werden. Damit kann man in logarithmischer Zeit von den Blättern zur Wurzel finden. Dort sind die benötigten Information gespeichert. Der Zugriff benötigt also wegen der logarithmischen Tiefe des Baumes logarithmische Zeit und damit sind die Operationen *setInfo* und *getInfo* in $O(\log n)$.

Es bleibt also noch die Operation *before(o₁, o₂)*. Dazu speichern wir bei jedem Knoten des Baumes der wie viele Sohn des Vaters er ist. Um nun *before(o₁, o₂)* zu berechnen, wird der Baum von den Blättern, die e_1 und e_2 entsprechen, zur Wurzel durchlaufen. Dabei wird ein Wort generiert, das sich aus der Sequenz der Nummern des jeweiligen Sohnes vom Blatt zur Wurzel zusammensetzt. Bei jedem Schritt in Richtung Wurzel wird die aktuelle Nummer vorne an das Wort angehängt.

Der Wert der Funktion *before* ergibt sich dann aus der lexikographischen Ordnung auf Wörtern: Sei $s[i]$ der i -te Eintrag eines Wortes. Sei $s_1 < s_2$ definiert durch $s_1 < s_2 \Leftrightarrow$

$\exists i : \forall j < i : s_1[j] = s_2[j] \wedge s_1[i] < s_2[i]$. Dann ist $\text{before}(e_1, e_2) = \text{true} \Leftrightarrow s_1 < s_2$, wobei s_i das e_i zugeordnete Wort ist. Damit ist auch diese Operation in $O(\log n)$ berechenbar und alle Operationen auf den Sektorbäumen liegen in $O(\log n)$. ■

Wir geben als nächstes alle für die Operationen benötigten Attribute und Datenstrukturen an. Danach zeigen wir, wie mit ihrer Hilfe die Einfügeoperationen in logarithmischer Zeit implementiert werden können.

5.3.2 Attribute von Knoten, Kanten und Flächen

Die folgende Liste zeigt alle verwendeten Attribute von Knoten, Kanten und Flächen. Attribute, die bereits bei der Implementierung mit linearer Laufzeit verwendet wurden und weiterhin benötigt werden, werden hier der Vollständigkeit halber erneut angegeben.

1. Attribute eines Knotens v :

Ein Knoten hatte bisher keine Attribute. Jetzt werden folgende benötigt:

- $sb[1], sb[2]$: Nach Lemma 5 ist jeder Knoten in maximal zwei Flächen Sektor­mittelknoten. $sb[1]$ und $sb[2]$ sind die Verweise in die beiden Sektorbäume, die den Sektoren, in denen v Sektormittelknoten ist, zugeordnet sind.
- $edge[1], edge[2]$: Zu jedem Verweis in einen Sektorbaum mit den Zeigern $sb[1]$ und $sb[2]$ existiert ein Verweis auf eine, diesem Verweis zugeordnete, Kante. Das ist die Kante, die v in der Fläche, der der Sektorbaum zugeordnet ist, im Uhrzeigersinn nachfolgt.

In Abbildung 5.8 sind alle Attribute eines Knotens v , der in zwei Flächen f_1 und f_2 Sektormittelknoten ist, dargestellt. Für einen Sektorstartknoten sind keine zusätzlichen Attribute nötig.

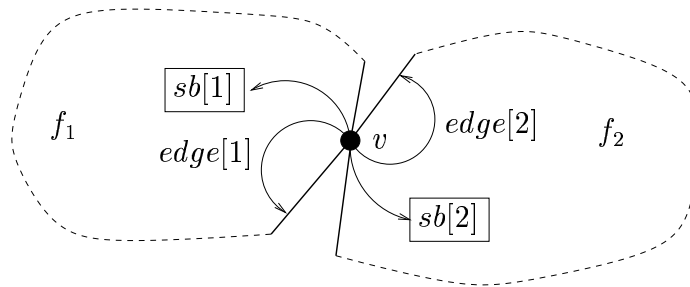


Abbildung 5.8: Knotenattribute.

2. Attribute einer Kante $e = \{v_1, v_2\}$:

Neben den bereits aus dem letzten Abschnitt bekannten Attributen kommt ein zusätzlicher Verweis auf ein Blatt in einem Sektorbaum hinzu.

- $(node1, node2)$: Die beiden Endknoten der Kante.
- \preceq_x und \preceq_y : Ordnung der Endknoten in x- und y-Richtung.
- $succ(v_1), pred(v_1), succ(v_2)$ und $pred(v_2)$: Doppelt verkettet, zyklische Verwaltung der Kanten um einen Knoten.
- $sb[1], sb[2]$: Die beiden Sektorbäume, die zu den Sektoren der angrenzenden Flächen gehören, in denen die Kante liegt.

Abbildung 5.9 zeigt einen Ausschnitt aus den Attributen einer Kante.

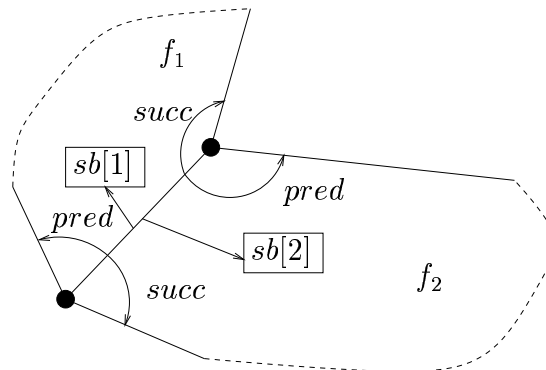


Abbildung 5.9: Kantenattribute

3. Attribute einer Fläche f :

Hier werden die alten Attribute $node$ und $edge$ aus dem letzten Kapitel nicht mehr benötigt. Statt dessen erhalten wir nun folgende Attribute:

- $sb[1], \dots, sb[4]$: Verweise auf die vier Sektorbäume, die den Sektoren der Fläche zugeordnet sind.
- $ssk[1], \dots, ssk[4]$: Verweise auf die vier Sektorstartknoten.
- $sskedge[1], \dots, sskedge[4]$: Die im Uhrzeigersinn erste Kante der Fläche, die zum jeweiligen Sektorstartknoten adjazent ist.

In Abbildung 5.10 sind die Attribute einer Fläche schematisch dargestellt.

Die Datenstrukturen und Attribute benötigen linearen Speicherplatz, denn ein planarer Graph enthält linear viele Objekte, die alle jeweils konstanten Speicherplatz benötigen. Außerdem benötigt auch die Menge aller Sektorbäume in Summe nur linear viel Speicherplatz. Denn 2-3 Bäume können mit linearem Speicherplatz implementiert werden ([AHU83]) und insgesamt werden nur linear viele Objekte, die Knoten und Kanten des Graphen, in den Bäumen repräsentiert.

Wir zeigen im Folgenden, wie mit Hilfe dieser Attribute und Datenstrukturen die Einfügeoperationen in logarithmischer Zeit implementiert werden können.

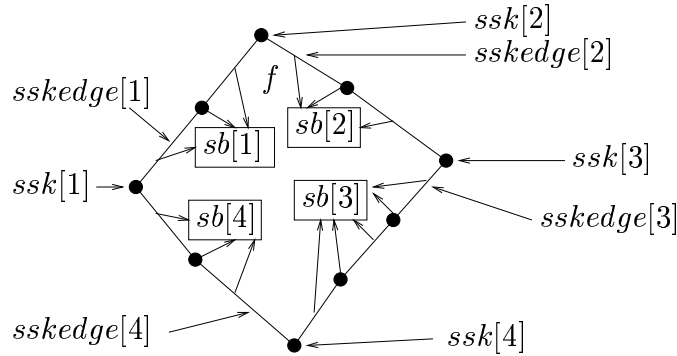


Abbildung 5.10: Attribute und Datenstrukturen einer Fläche

5.3.3 Einfügen eines Knotens

Der folgende Satz zeigt, dass ein Knoten in logarithmischer Zeit eingefügt werden kann.

Satz 5

Gegeben sei ein Graph mit einem legalen Restriktionensystem. Die Funktion *insertNode* aus Definition 8 kann in logarithmischer Laufzeit mit linearem Speicherbedarf implementiert werden.

BEWEIS Gegeben sei wieder die Funktion *insertNode* aus Definition 8 zum Einfügen eines neuen Knotens. Die Funktion ersetzt die Kante $e = \{v_1, v_2\}$ durch den neuen Knoten v und zwei neue Kanten $e_1 = \{v_1, v\}$ und $e_2 = \{v, v_2\}$.

Wir verwenden die oben beschriebenen Datenstrukturen. Die Attribute von Knoten, Kanten und Flächen werden wie folgt neu berechnet:

1. Attribute der Knoten v , v_1 und v_2 :

- $sb[1]$, $sb[2]$: Die für den neuen Knoten v zu verwendenden Sektorbäume sind die gleichen wie die der Kante e . Wir fügen den neuen Knoten v in die gleichen Sektorbäume ein, in denen auch die Kante e vor der Operation liegt. Wir fügen v hinter e ein, so dass v korrekterweise zwischen v_1 und v_2 liegt. Außerdem wird e später noch entfernt werden.
- $edge[1]$, $edge[2]$: Dem Knoten v folgen in einer der beiden Flächen die neue Kante e_1 und in der anderen e_2 . Es muss noch festgestellt werden, welche der beiden Kanten an $edge[1]$ und welche an $edge[2]$ zugewiesen wird. Wenn man die Einfügeoperation genauer betrachtet, dann sieht man, dass die Kante e_2 dem Knoten v in der Fläche folgt, in der e dem Knoten v_1 gefolgt ist. Diese Informationen lassen sich aus $e.sb[i].getInfo()$, $v_1.sb[i].getInfo()$ und $v_1.edge[i]$ berechnen. Mit $sb[i].getInfo()$ kann jeweils auf die Fläche eines Knotens oder einer Kante in einer Fläche zugegriffen werden.

Entsprechend der hier gefundenen Zuordnung kann dann auch e_2 einem der Attribute $v.edge[1]$ und $v.edge[2]$ zugeordnet werden.

Außerdem muss für v_1 und v_2 in den Flächen, in denen e Nachfolger der Knoten war, die Kante e durch e_1 bzw. e_2 in den Attributen $edge[i]$ ersetzt werden. Dies geschieht entweder in den Knoten- oder in den Flächenattributen, je nachdem, ob die Knoten in der entsprechenden Fläche Sektorstartknoten sind oder nicht.

Dazu vergleichen wir die Attribute $edge[i]$ der beiden Knoten v_1 und v_2 mit e . Wenn hier ein Verweis auf e gespeichert ist, dann wird das bei v_1 durch e_1 und bei v_2 durch e_2 ersetzt. Außerdem bestimmen wir mit $e.sb[i].getInfo()$ die beiden Flächen, in denen e liegt, und ersetzen nach dem gleichen Muster in den Attributen $f.sskedge[i]$ die Kante e durch e_1 (wenn v_1 Sektorstartknoten i ist) oder durch e_2 (wenn v_2 Sektorstartknoten i ist).

2. Attribute der Kanten e_1 und e_2 :

Für die Attribute $node[i]$ sind jeweils die beiden Endknoten zu verwenden. Die Ordnungsrelationen \preceq_x und \preceq_y können jeweils aus der Belegung der gleichen Attribute bei Kante e berechnet werden.

Die zyklische Ordnung der Kanten um die Knoten mit $succ()$ und $pred()$ erhalten wir, in dem e_1 und e_2 beim neuen Knoten v sich jeweils gegenseitig referenzieren und indem e bei v_1 durch e_1 und bei v_2 durch e_2 ersetzt wird. Die Kanten, bei denen e in den Attributen $succ()$ und $pred()$ ersetzt werden muss, werden mit Hilfe der gleichen Attribute bei e bestimmt.

Schließlich muss noch die Belegung der Attribute $sb[i]$ bestimmt werden. Die korrekten Sektorbäume können mit Hilfe der Attribute $sb[i]$ von Knoten v berechnet werden. Allerdings muss darauf geachtet werden, ob die Kanten vor oder nach v eingefügt werden müssen. Die dazu notwendigen Informationen erhält man aus der Information, welche der beiden Kanten v jeweils in der Fläche $sb[i].face()$ folgt (Attribut $v.edge[i]$).

Zum Abschluss muss noch e aus den beiden Sektorbäumen entfernt werden.

3. Attribute der Fläche f :

Die Flächenattribute müssen nicht mehr weiter geändert werden. Alle nötigen Änderungen an den Sektorbäumen wurden bereits durch das Einfügen der neuen Objekte und das Entfernen der Kante e durchgeführt. Die Sektorstartknoten können sich nicht geändert haben, lediglich die Kanten, die den Sektorstartknoten folgen. Diese Änderung wurde aber bereits bei der Änderung der Knotenattribute angegeben.

Zur Implementierung benötigen wir neben dem direkten Zugriff auf Knoten-, Kanten- und Flächenattribute auch noch die Funktionen $insert()$ und $delete()$ eines Sektorbaumes. Diese Funktionen können in $O(\log n)$ implementiert werden.

Alle Objekte haben nur konstanten Speicherbedarf, außer die Sektorbäume. Bäume benötigen linearen Speicher in der Anzahl der darin gespeicherten Objekte. In Summe

sind bei uns linear viele Objekte gespeichert. In Summe erhalten wir also linear viel Speicherplatz. ■

5.3.4 Einfügen einer Kante

Der folgende Satz zeigt, dass die Datenstrukturen auch das Einfügen einer Kante in logarithmischer Zeit erlaubt.

Satz 6

Gegeben sei ein Graph mit einem legalen Restriktionensystem. Die Funktion *insertEdge* aus Definition 9 kann in logarithmischer Laufzeit mit linearem Speicherbedarf implementiert werden.

BEWEIS Gegeben sei die Funktion *insertEdge* aus Definition 9, die zwischen den Knoten v_1 und v_2 in Fläche f die neue Kante $e = \{v_1, v_2\}$ einfügt. Die Fläche f wird dabei in zwei neue Flächen f_1 und f_2 gespalten.

Wir zeigen, wie die Attribute von Knoten, Kanten und Flächen neu berechnet werden. Dazu legen wir zunächst die Zuordnung der beiden neuen Flächen f_1 und f_2 fest. Sei f_1 die Fläche, die die Knoten von v_1 bis v_2 und die Kante e im Uhrzeigersinn um f enthält. Die Fläche f_2 enthält damit die Knoten von v_2 bis v_1 und die Kante e . Natürlich gehören dann auch jeweils die Kanten zwischen den Knoten zu den Flächen (vgl. Abbildung 5.5).

Sei außerdem S_1 der Sektor von v_1 und S_2 der Sektor von v_2 in der Fläche f . Wir beginnen die Beschreibung der Neuberechnung der Attribute mit den Flächen, da alle anderen Änderungen davon abhängen.

1. Attribute der Flächen f_1 und f_2 :

Wir müssen für die beiden neuen Flächen f_1 und f_2 die Attribute neu berechnen. Dabei dürfen wir die Attribute von f verwenden und zerstören, da sie im Folgenden nicht mehr benötigt werden.

- $sb[1], \dots, sb[4]$: Alle Sektoren, die in der Fläche f zwischen Sektor S_1 und S_2 liegen, können unverändert in Fläche f_1 übernommen werden. Ebenso können alle Sektoren, die zwischen Sektor S_2 und S_1 liegen, unverändert in Fläche f_2 weiterverwendet werden.

Die Sektoren S_1 und S_2 finden sich nach der Einfügeoperation zum Teil in der Fläche f_1 und zum Teil in der Fläche f_2 . Deshalb müssen sie zuerst bei den Knoten v_1 und v_2 auseinandergeschnitten werden und die entstehenden Teile dann entsprechend an f_1 und f_2 zugewiesen werden. Das geschieht mit Hilfe der Funktion *split* eines Sektorbaumes.

Alle Verweise auf Sektorbäume in den Flächen f_1 und f_2 , denen wir bisher keine Objekte zugewiesen haben, werden nun mit einem leeren Sektorbaum initialisiert. Bei sämtlichen Sektorbäumen beider Flächen wird der Name der jeweiligen Fläche noch gespeichert.

- $ssk[1], \dots, ssk[4], sskedge[1], \dots, sskedge[4]$: Die Sektorstartknoten in den beiden Flächen f_1 und f_2 setzen sich aus den Sektorstartknoten der Fläche f und den Knoten v_1 und v_2 zusammen.

Die Knoten v_1 und v_2 werden je nach der Belegung der Attribute \preceq_x und \preceq_y der neuen Kante e zu Sektorstartknoten. Ob sie ein Sektorstartknoten sind, und wenn ja welcher, kann leicht durch \preceq_x und \preceq_y der Kanten, die an die Knoten in der jeweiligen Fläche anschließen, bestimmt werden.

Entsprechend den Sektorstartknoten sind auch die Attribute $sskedge[1], \dots, sskedge[4]$ zu belegen. Betrachten wir dazu Sektorstartknoten j in Fläche f_1 oder f_2 . Wenn der Knoten weder v_1 noch v_2 ist, dann war er bereits in Fläche f Sektorstartknoten j und die Belegung des Attributs $sskedge[j]$ in der neuen Fläche kann aus dem Attribut $sskedge[j]$ der Fläche f übernommen werden.

Wir müssen noch die Fälle bearbeiten, in denen v_1 oder v_2 ein Sektorstartknoten in den neuen Flächen ist. Der Nachfolger von diesen beiden Knoten ist entweder der Nachfolger des Knotens in Fläche f oder die neue Kante e .

Wegen der Wahl von v_1 und v_2 und der Festlegung, welche Fläche f_1 und f_2 ist muss die Kante e der Nachfolger von v_1 in f_2 und von v_2 in f_1 sein. Falls e nicht der Nachfolger ist, dann kann die Nachfolgerkante aus den Knotenattributen berechnet werden. Hier ist die Kante, die einem Knoten in einer Fläche folgt, abgelegt.

2. Attribute der Knoten v_1 und v_2 :

Bei allen Knoten in Fläche f hat sich die Fläche, in der sie liegen, geändert. Sie liegen nicht mehr in f , sondern in Fläche f_1 oder f_2 . Die Knoten haben aber keine direkten Verweise auf die Fläche, sondern nur indirekte über die Sektorbäume. Bei diesen wurde aber die Fläche schon in den Sektorbäumen abgepasst.

Wir müssen also lediglich für die Knoten v_1 und v_2 die Attribute neu berechnen.

- $sb[1], sb[2]$: Falls ein Knoten in einer der beiden Flächen Sektorstartknoten ist, dann muss er, vorausgesetzt er ist noch Element eines Sektorbaumes, aus diesem gelöscht werden. Das ist nötig, weil ein Sektormittelknoten in Fläche f zu einem Sektorstartknoten in Fläche f_1 und/oder f_2 werden kann.

Falls einer der beiden Knoten in Fläche f_i kein Sektorstartknoten ist und in f aber einer war, dann muss er noch in den passenden Sektorbaum eingetragen werden. Dazu muss die Kante bestimmt werden, hinter der der Knoten einzufügen ist. Dies ist entweder die Vorgängerkante in f oder aber die neue Kante e , abhängig davon ob wir f_1 oder f_2 betrachten.

Da wir hierfür voraussetzen, dass die Kante e in den Sektorbäumen repräsentiert ist, wird diese Funktion erst nach dem Belegen der Kantenattribute von e durchgeführt.

- $edge[1], edge[2]$: Die Kante e wird in der Fläche f_2 Nachfolger von v_1 und in Fläche f_1 Nachfolger von v_2 . Wir müssen diese Kante aber nur dann bei den Knotenattributen eintragen, wenn v kein Sektorstartknoten ist.

3. Attribute der neuen Kante e :

Die Attribute $node[i]$ müssen nur für die Kante e festgelegt werden. Bei allen anderen Kanten hat sich nichts geändert. Die Berechnung der Lageabhängigkeiten \preceq_x und \preceq_y erfolgt entsprechend Definition 9.

Die Anordnung der Kanten um die Endknoten muss bei v_1 und v_2 ergänzt werden. Die Kante e wird dazu zwischen den beiden Kanten, die sich in Fläche f an Knoten v_i angeschlossen haben, eingefügt. Die erste Kante kann mit Hilfe von $v.edge()$ bzw. $f.sskedge()$ gefunden werden und die zweite mit $succ$.

Es fehlen noch die Attribute $sb[1]$ und $sb[2]$. Die Kante e wird in den beiden Flächen f_1 und f_2 in die Sektorbäume gemäß dem in Definition 9 festgelegten Sektoren eingefügt. Der Vorgänger ist dabei in Fläche f_1 der Knoten v_2 und in f_2 der Knoten v_1 . Falls der entsprechende Knoten in der Fläche Sektorstartknoten ist, dann ist e das erste Element des Sektors.

Damit sind alle Datenstrukturen wiederhergestellt. Wir benötigen zum Implementieren neben dem Zugriff auf direkt vorhandene Attribute von Knoten, Kanten oder Flächen noch die Funktionen $face()$, $insert()$, $split()$ und $delete()$ eines Sektorbaumes. Diese Funktionen können, wie in Abschnitt 5.3.1 gezeigt, in logarithmischer Zeit implementiert werden.

Der Speicherbedarf ist linear. Denn es werden die gleichen Datenstrukturen wie beim Einfügen eines Knotens verwendet und in Satz 5 wurde gezeigt, dass linearer Speicher genügt. ■

Wir können also mit einer relativ einfachen Datenstruktur die Einfügeoperationen in logarithmischer Zeit implementieren. Den Schlüssel dazu gibt uns Lemma 5. Mit diesem Lemma können wir die Anzahl der Attribute, die wir bei jedem Objekt speichern müssen, auf konstante Größe reduzieren.

Auch die Implementierung der Funktionen ist relativ einfach. Sie enthält allerdings, insbesondere für das Einfügen einer neuen Kante, relativ viele Fallunterscheidungen. Die vielen Fälle ergeben sich auf Grund der Tatsache, dass wir nach den Sektoren der Endknoten unterscheiden müssen. Zusätzlich muss auch noch berücksichtigt werden, ob die Knoten Sektorstartknoten sind oder nicht. Die Ideen bleiben aber einfach und sind überall identisch.

6 Erzeugung einer orthogonalen Repräsentation

Im letzten Kapitel haben wir gezeigt, wie aus einem initialen Graph mit einem Restriktionensystem mit Hilfe von Einfügeoperationen Schritt für Schritt ein Graph $G = (V, E, R)$ aufgebaut wird.

Um unser Ziel, einen interaktiven Zeichenalgorithmus, zu erreichen, muss nun noch für $G = (V, E, R)$ eine Zeichnung erzeugt werden. Dabei können wir von der Voraussetzung ausgehen, dass das Restriktionensystem dem 4-Sektorenmodell entspricht. Das gilt, weil es eine Invariante unserer Einfügeoperationen ist, dass stets das Restriktionensystem dem 4-Sektorenmodell entspricht.

Eine beim orthogonalen Zeichnen von Graphen etablierte Vorstufe zu einer Zeichnung ist eine orthogonale Repräsentation des Graphen. Es existieren diverse Standard-Algorithmen, die aus einer orthogonale Repräsentation eine orthogonale Zeichnung des Graphen erzeugen (siehe z.B. [FK95] für einen Algorithmus für das Kandinsky-Modell).

Auch wir wählen hier den Weg über die orthogonale Repräsentation. Das heißt, zum Zeichnen des Graphen müssen wir angeben, wie für einen Graphen mit einem legalen Restriktionensystem eine orthogonale Repräsentation erzeugt werden kann.

Die Einbettung ist bereits Teil des Restriktionensystems. Damit müssen zusätzlich noch die Kantenverläufe festgelegt werden. Außerdem ist zu zeigen, dass es für die konstruierte orthogonale Repräsentation eine entsprechende orthogonale, planare Zeichnung gibt.

Eine übliche Beschreibung der Kantenverläufe in einer orthogonale Repräsentation listet für jede Kante die Liste aller Links- und Rechtsknicken auf. Zusätzlich wird bei jedem Knoten zwischen je zwei aufeinander folgenden Kanten beschrieben, in welchem Winkel (0, 90, 180 oder 270 Grad) die beiden Kanten aufeinander stehen. Wie wir sehen werden, gibt es in unserer Anwendung nur vier mögliche Kantenverläufe, so dass uns eine einfachere Beschreibung der Kantenverläufe genügt. Wir geben jeweils nur an, von welchem Typ der Verlauf einer Kante ist. In Abbildung 6.1 sind diese verschiedenen Alternativen für die Kantenverläufe dargestellt, sie werden in diesem Kapitel, gemeinsam mit ihrer Entstehung, noch erklärt. Aus diesen Typen kann die übliche Notation leicht abgeleitet werden, so dass dieser Schritt hier nicht beschrieben wird. Wenn wir uns im nächsten Kapitel noch mit der Einsparung von Knicken beschäftigen, dann kommen noch vier weitere Kantenverläufe hinzu.

Die Eingabe ist in diesem Kapitel der Graph und das Restriktionensystem und als

Ausgabe wird eine orthogonale Repräsentation generiert.

Wie sehen die verschiedenen möglichen Kantenverläufe aus, und warum haben wir genau diese Alternativen gewählt?

Hauptgesichtspunkt bei einem interaktiven Algorithmus sollte sein, dass sich zwei aufeinander folgende Zeichnungen — auch bei Änderungen des Graphen durch eine Einfügeoperation — möglichst stark ähneln. Zu diesem Zweck werden zwei Gesichtspunkte besonders berücksichtigt:

1. Die Zeichnung erhält eine strenge Orientierung von links nach rechts, genauso wie man es vom Lesen gewohnt ist. Dazu beginnt jede Kante immer auf der rechten Seite des linken Endknotens. Nach einem horizontalen Kantenstück folgt nach einem Knick ein vertikales Kantenstück. Die Richtung des Knickes ergibt sich aus den Lagebeziehungen der Endknoten. Wenn der linke Endknoten unterhalb des rechten liegt, dann handelt es sich um einen Linksknick, andernfalls um einen Rechtsknick. Falls die Kante zwei Knicke hat, folgt ein weiteres horizontales Kantenstück und die Kante endet auf der linken Seite des rechten Knotens.
2. Kantenverläufe werden nur dann geändert, wenn dies zur Erhaltung der Planarität unbedingt notwendig ist. Das heißt, jede Kante wird wenn möglich mit einem Knick gezeichnet. Die genauen Regeln hierfür werden in diesem Kapitel noch angegeben. Nur wenn mit einem Knick keine planare Zeichnung erzeugt werden kann, erhält eine Kante zwei Knicke.

In [BT98] wird explizit gefordert, dass sich Kantenverläufe bei interaktiven Algorithmen so wenig wie möglich ändern.

Theoretisch gäbe es noch die Möglichkeit, alle Kanten immer mit zwei Knicken zu zeichnen. Dann würde sich ein Kantenverlauf nie ändern. Unserer Meinung nach hat eine Zeichnung dann aber zu viele Kantenknicke und wird damit unübersichtlich.

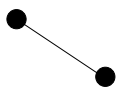
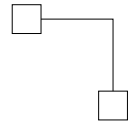
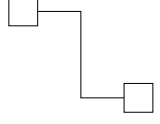
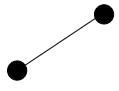
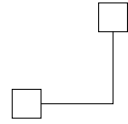
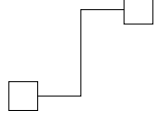
Lagebeziehungen	ein Knick	zwei Knicke
	 Kantenverlauf 1	 Kantenverlauf 2
	 Kantenverlauf 3	 Kantenverlauf 4

Abbildung 6.1: Erzeugte Kantenverläufe.

Als Ergebnis werden wir einen Zeichenalgorithmus erhalten, der folgende Anforderungen erfüllt:

1. Die erzeugte Zeichnung respektiert das Restriktionensystem.
2. Die Laufzeit des Algorithmus ist $O(n)$.
3. Maximal $|F|$ Kanten haben zwei Knicke. Alle anderen Kanten haben einen Knick.
4. Der Verlauf einer Kante ändert sich nicht, solange die Anzahl an Knicken in der Kante nicht geändert wird.

Um die Anforderungen zu beweisen benötigen wir zunächst noch ein Lemma. Wir betrachten in diesem Lemma zwei Knoten in einer gemeinsamen Fläche f , die nicht unbedingt durch eine Kante verbunden sein müssen. Auf Grund der Transitivität der Lagebeziehungen zwischen je zwei adjazenten Knoten soll zwischen den beiden Knoten eine vorgeschriebene Lagebeziehung in x- oder y-Richtung gelten. Wir zeigen, dass alleine wegen der Lagebeziehungen von Knoten und Kanten in der Fläche f ebenfalls die Lagebeziehung zwischen den beiden Knoten erzwungen wird.

Lemma 6

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Seien u und v zwei Knoten, die in einer gemeinsamen Fläche f liegen. Wenn die beiden Knoten auf Grund der Attribute \preceq_x und \preceq_y aller Kanten in x- oder y-Richtung geordnet sind, dann sind sie auch alleine wegen der Kanten in Fläche f in x- oder y-Richtung geordnet.

BEWEIS Wir zeigen die Behauptung nur für eine Ordnung in x-Richtung. Der Beweis für eine Ordnung in y-Richtung ist entsprechend.

Gelte also, dass u links von v liegt. Damit die Knoten in x-Richtung geordnet sind, muss es eine Sequenz von Knoten v_1, \dots, v_k mit $u = v_1$ und $v = v_k$ geben, so dass $\{v_i, v_{i+1}\} \in E$ und $\{v_i, v_{i+1}\} \preceq_x = t$. Damit muss v_i links von v_{i+1} liegen. Diese Knoten und Kanten bilden einen Pfad von u nach v .

Da der Pfad die Fläche f nicht schneiden kann, liegt f entweder ober- oder unterhalb des Pfades. Nehmen wir zunächst an, dass f unterhalb des Pfades liegt. Wegen der gegebenen Ordnung muss gelten, dass die erste Kante des Pfades Knoten u nach rechts verlassen muss. Ebenso gilt, dass die letzte Kante an Knoten v von links anschließen muss.

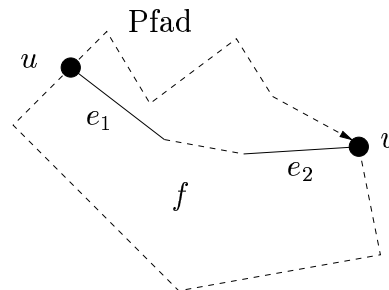


Abbildung 6.2: Bezeichnungen zum Beweis von Lemma 6.

Wir betrachten die in zyklischer Reihenfolge erste Kante e_1 , die an Knoten u in Fläche f anschließt (siehe Abbildung 6.2). Außerdem sei e_2 die erste Kante im Gegenuhrzeigersinn, die in Fläche f liegt und zu v adjazent ist.

Die Kante e_1 kann u nicht nach links verlassen, da sonst in Fläche f das 4-Sektorenmodell verletzt wäre. Denn dann müsste auch die zweite Kante, die zu Knoten u in Fläche f adjazent ist, den Knoten nach links verlassen. Dann müssten wegen des 4-Sektorenmodells alle anderen Knoten und Kanten der Fläche links von u liegen, was ein Widerspruch dazu ist, dass v rechts von u liegt. Da die Kante e_1 den Knoten u nach rechts verlässt, liegt u und damit e_1 in der Fläche f in Sektor 1 oder Sektor 2.

Ebenso muss die Kante e_2 den Knoten v nach links verlassen. Deshalb liegt e_2 in der Fläche f ebenfalls in Sektor 1 oder Sektor 2. Wegen des 4-Sektorenmodells müssen dann aber alle anderen Kanten zwischen e_1 und e_2 ebenfalls in diesen beiden Sektoren liegen und damit existiert in der Fläche f ein Pfad von u zu v , so dass u und v auf Grund des Pfades in x -Richtung geordnet sind.

Der Fall, dass der Pfad unter der Fläche f liegt, ist symmetrisch. ■

Das folgende Lemma zeigt, wie die Kantenverläufe festgelegt werden und welche Kanten zwei Knicke haben müssen. Zuvor wollen wir aber noch an Beispielen erklären, warum und wann Kanten mit zwei Knicken notwendig sind. Wir beschränken uns dabei auf zwei von vier verschiedenen Möglichkeiten, da die beiden anderen durch eine Spiegelung an der x -Achse aus ihnen hervorgehen.

Betrachten wir zunächst Abbildung 6.3(a). Hier muss der linke Endknoten v_1 der Kante e links oberhalb des rechten Endknotens v_2 liegen. Und es gibt (mindestens) einen Knoten, der links oberhalb von v_2 und rechts unterhalb von v_1 liegen muss. Die Kante e muss auf Grund der Einbettung unterhalb dieser Knoten verlaufen. Mit der Einschränkung, dass eine Kante immer auf der rechten Seite des linken Endknotens beginnt, ist das in einer planaren Zeichnung nur dann möglich, wenn die Kante e zwei Knicke hat. Dieses Beispiel wird durch die Bedingung eins im nachfolgenden Lemma 7 charakterisiert und in der Spiegelung durch die Bedingung 2.

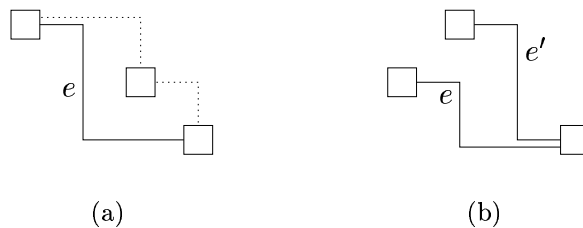


Abbildung 6.3: Fälle, in denen Kanten mit zwei Knicken gezeichnet werden müssen.

In Abbildung 6.3(b) hat der zyklische Nachfolger e' von e am rechten Endknoten zwei Knicke. Beide Kanten müssen auf Grund der Lagebeziehungen den Knoten nach links oben verlassen. Wenn e einen Knick hätte, dann müsste die Kante auf der Oberseite vom

rechten Endknoten enden. Allerdings ist das nicht möglich, da e' diesen Weg versperrt. Dieses Beispiel wird durch die Bedingung drei in Lemma 7 charakterisiert und in der Spiegelung durch die Bedingung vier.

Wir beziehen uns im folgenden Lemma bei der Angabe der Kantenverläufe auf Abbildung 6.1.

Lemma 7 (Bestimmung der orthogonalen Repräsentation)

Gegeben sei ein Graph $G = (V, E, R)$ mit einem legalen Restriktionensystem R .

Dann wird wie folgt eine orthogonale Repräsentation konstruiert:

Jede Kante beginnt auf der rechten Seite des linken Endknotens. Nach einem waagerechten Kantenstück folgt ein senkrechtes, wobei sich der Knick aus den Lagebeziehungen der beiden Endknoten ergibt. Wenn der linke Endknoten unterhalb des rechten liegt, dann handelt es sich um einen Linksknick (Kantenverlauf 3 oder 4 in Abbildung 6.1), andernfalls um einen Rechtsknick (Kantenverlauf 1 oder 2 in Abbildung 6.1).

Eine Kante e erhält nach dem waagerechten Kantenstück einen zweiten Knick und ein zusätzliches waagerechtes Kantenstück, wenn eine Fläche f existiert, in der eine der folgenden vier Eigenschaften erfüllt ist:

- 1. Sektor 1 und Sektor 3 sind leer und Sektor 4 besteht nur aus der Kante e . Dann wird der Kantenverlauf 2 aus Abbildung 6.1 verwendet.*
- 2. Sektor 2 und Sektor 4 sind leer und Sektor 1 besteht nur aus der Kante e . Dann wird der Kantenverlauf 4 aus Abbildung 6.1 verwendet.*
- 3. Der zyklische Nachfolger von e am rechten Endknoten v hat zwei Knicke und beide Kanten verlassen den Knoten v nach links oben. Dann wird der Kantenverlauf 2 aus Abbildung 6.1 verwendet.*
- 4. Der zyklische Vorgänger von e am rechten Endknoten v hat zwei Knicke und beide Kanten verlassen den Knoten v nach links unten. Dann wird der Kantenverlauf 4 aus Abbildung 6.1 verwendet.*

Falls keine dieser Eigenschaften erfüllt ist, dann hat die Kante einen Knick und endet nach dem senkrechten Kantenstück am rechten Endknoten.

Für die so konstruierte orthogonale Repräsentation existiert eine entsprechende orthogonale planare Zeichnung im Kandinsky-Modell.

BEWEIS Wir zeigen, dass jede Fläche planar gezeichnet werden kann. Damit kann auch der Gesamtgraphen planar gezeichnet werden.

Zunächst gilt, dass sich Knoten und Kanten innerhalb eines Sektors nicht schneiden können, da sie auf Grund der Lagebeziehungen sowohl in x- als auch in y-Richtung geometrisch getrennt sein müssen.

Ebenso können sich die Knoten und Kanten eines Sektors nicht mit dem Sektorstartknoten des folgenden Sektors und der Kante zu ihm schneiden, da auch dieser Knoten auf Grund der Lagebeziehungen geometrisch getrennt liegen muss.

Als nächstes zeigen wir, dass eine Zeichnung existiert, in der sich Knoten und Kanten aus verschiedenen Sektoren nicht schneiden können, außer sie enden an einem gemeinsamen Knoten.

Man kann jede Fläche so zeichnen, dass die Knoten der verschiedenen Sektoren, und damit die Kanten zwischen ihnen, voneinander getrennt liegen (vgl. Abbildung 6.4). Wir ordnen die Knoten der Fläche dazu so an, dass Sektormittelknoten in Sektor 1 und 2 von Sektor 3 und 4 durch eine waagerechte Linie, und Sektor 1 und 4 von Sektor 2 und 3 durch eine senkrechte Linie getrennt werden. Für die Sektorstartknoten kann das nicht erfüllt werden (Abbildung 6.4(b) zeigt ein Beispiel, in dem das nicht möglich ist), wenn ein Sektor leer ist. In diesem Fall erzwingen die Lagebeziehungen zwischen den Knoten, dass der Sektorstartknoten nicht wie gewünscht platziert werden kann. Allerdings haben wir schon oben gesehen, dass sich die Sektorstartknoten nicht mit Knoten und Kanten des Vorgängersektors schneiden können.

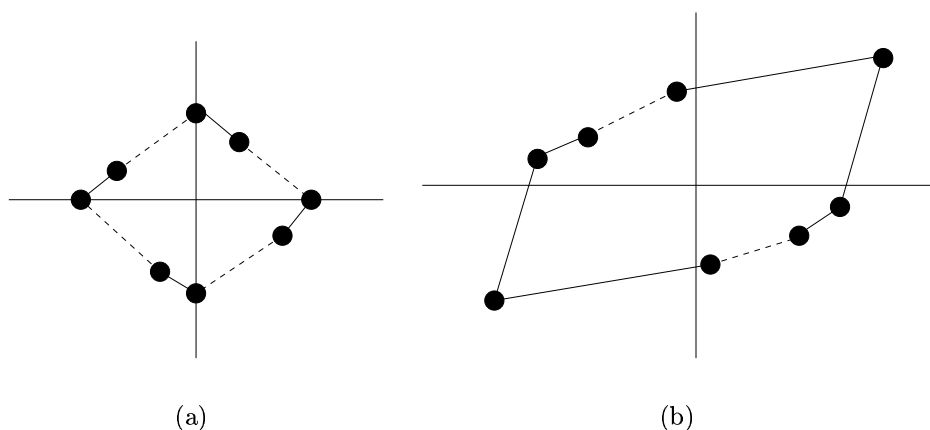


Abbildung 6.4: Trennung der Sektoren.

Damit muss nur noch untersucht werden, ob sich die Kanten, die in verschiedenen Sektoren liegen und einen gemeinsamen Endknoten haben, schneiden können. Folglich müssen noch die Kanten in einer Fläche betrachtet werden, die an einem Sektorstartknoten aufeinander treffen.

Wir betrachten zunächst die rechte Seite eines Knotens v . Sei $e_1 = \{v, v_1\}$ eine Kante die an der rechten Seite des Knotens endet. o. B. d. A gelte $d(e_1, v) = \nearrow$.¹ Sei $e_2 = \{v, v_2\}$ der Vorgänger von e_1 am Knoten v . Wir zeigen, dass sich die beiden Kanten nicht schneiden können.

Falls e_2 nicht auf der rechten Seite von v endet, können sich die Kanten nicht schneiden. Denn in diesem Fall liegt v_1 und damit die Kante e_1 rechts von v , wogegen v_2 und e_2 links und unter- oder oberhalb von v liegen.

¹ Falls $d(e_1, v) = \nwarrow$ gilt, dann müssen nur die folgenden Überlegungen an der x-Achse gespiegelt werden.

Nehmen wir also an, dass die beiden Kanten auf der rechten Seite von v enden. Dann kann es keinen Kantenschnitt geben, wenn die Kante e_2 den Knoten v nach rechts oben verlässt. Denn dann liegt e_1 unterhalb und e_2 oberhalb von v .

Wenn beide Kanten den Knoten v nach rechts unten verlassen, dann muss der senkrechte Teil der Kante e_2 rechts vom senkrechten Teil von e_1 liegen. Wenn es im Restriktionensystem keine Lageabhängigkeit gibt, die festlegt, dass v_2 links von v_1 liegen muss, dann existiert eine Zeichnung, in der sich die beiden Kanten nicht schneiden. Dazu wird v_2 einfach rechts von v_1 gezeichnet.

Nehmen wir also an, dass es im Restriktionensystem eine Lageabhängigkeit gibt, die dafür sorgt, dass v_2 links von v_1 liegen muss. Wir zeigen, dass dann e_1 zwei Knicke haben muss und damit kann der senkrechte Teil von e_1 wieder links vom senkrechten Teil von e_2 liegen. Betrachten wir dazu die Fläche f , die von e_1 und e_2 begrenzt wird (siehe Abbildung 6.5).

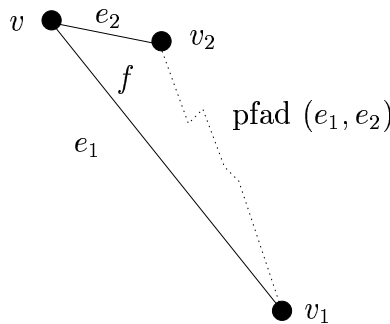


Abbildung 6.5: Bezeichnungen zum Beweis von Lemma 7.

Da v_2 links von v_1 liegen muss, gibt es im Restriktionensystem einen Pfad von v_2 zu v_1 , der erzwingt, dass v_2 und v_1 auf Grund dieses Pfades in x-Richtung sortiert sind. Wenn so ein Pfad existiert, dann muss er auch in f existieren (vgl. Lemma 6). Dazu müssen aber alle Knoten und Kanten des Pfades in Fläche f in Sektor 1 oder 2 liegen. Da e_2 schon in Sektor 2 liegt, liegen sie sogar alle in Sektor 2.

Dann existiert aber eine Fläche (die Fläche f), in der die Knoten und Kanten nur in Sektor 2 und 4 liegen. Zusätzlich besteht Sektor 4 nur aus e_1 . Somit muss e_1 nach Bedingung 1 zwei Knicke haben.

Als nächstes betrachten wir Kanten, die auf der linken Seite des Knotens enden. Eine Kante endet nur dann auf der linken Seite eines Knotens, wenn sie zwei Knicke hat. Sei $e_1 = \{v_1, v\}$ eine Kante mit zwei Knicken, die auf der linken Seite von Knoten v endet und o. B. d. A sei $d(e_1, v) = \swarrow$. Jede andere Kante, die auf der linken Seite von v endet, hat ebenfalls zwei Knicke. Da aber die senkrechten Teile von Kanten mit zwei Knicken nicht gegeneinander geordnet sind, können sie wieder so angeordnet werden, dass kein Kantenschnitt entsteht.

Die Kante e_1 kann sich auch mit keiner Kante $e_2 = \{v_2, v\}$ schneiden, die auf der Oberseite von v endet. Denn dazu müsste es Lageabhängigkeiten geben, die dafür sorgen, dass v_1 rechts oberhalb von v_2 liegen muss. Diese Abhängigkeiten muss es dann auch in

der Fläche f geben, die von e_1 und e_2 begrenzt wird (Lemma 6). Dann entspricht aber f nicht dem 4-Sektorenmodell, da dann bei v die Einbettung keine planare Zeichnung erlaubt.

Schließlich betrachten wir die Oberseite von Knoten v . Seien $e_1 = \{v, v_1\}$ und $e_2 = \{v, v_2\}$ zwei Kanten, die auf der Oberseite von v enden und sei e_1 der zyklische Vorgänger von e_2 im Uhrzeigersinn. Da die beiden Kanten auf der Oberseite von v enden, gilt $d(e_1, v) = d(e_2, v) = \searrow$ und beide Kanten haben einen Knick. Das waagerechte Kantenstück von e_1 muss unterhalb dem waagerechten Kantenstück von e_2 liegen.

Angenommen, das ist nicht möglich. Dann muss in der Fläche, die von e_1 und e_2 begrenzt wird, ein Pfad von v_1 zu v_2 existieren, so dass v_1 oberhalb von v_2 liegen muss. Dann liegt aber in f die Kante e_1 in Sektor 4 und alle anderen Knoten und Kanten in Sektor 2. Damit muss e_1 zwei Knicke haben und auf der linken Seite von v enden. ■

Damit haben wir gezeigt, wie aus einem Restriktionensystem eine orthogonale Repräsentation generiert werden kann. In Abbildung 6.6 ist als Beispiel ein Restriktionensystem und die daraus konstruierte orthogonale Repräsentation dargestellt. Wie immer entspricht die Darstellung der orthogonalen Repräsentation schon einer orthogonalen Zeichnung.

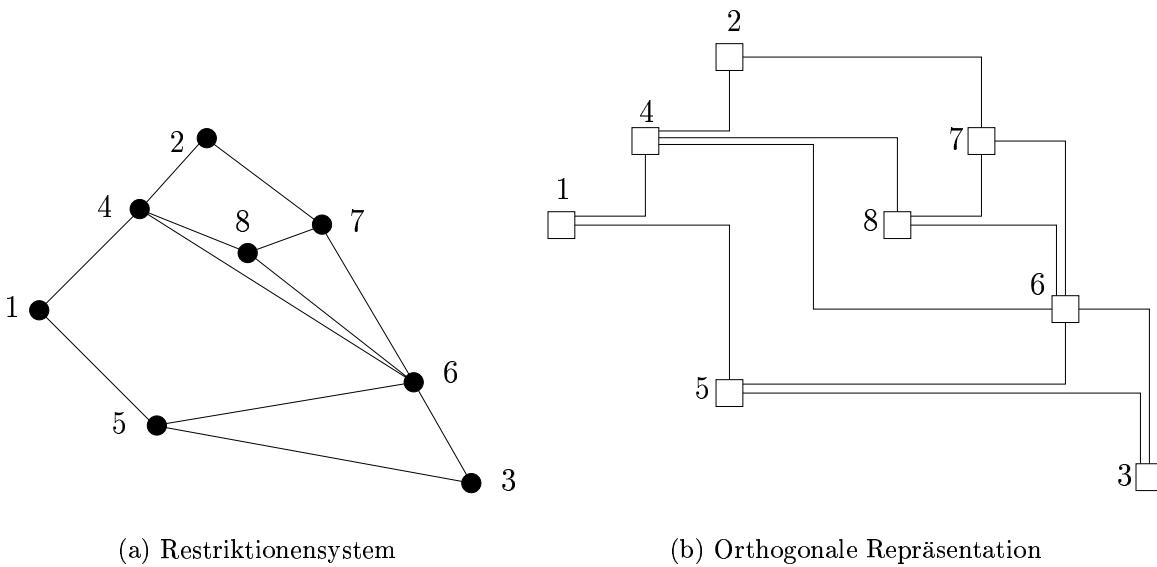


Abbildung 6.6: Restriktionensystem und orthogonale Repräsentation.

Wir zeigen die übrigen oben postulierten Eigenschaften.

Lemma 8 (Die orthogonale Repräsentation kann in $O(n)$ berechnet werden)

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Dann kann in $O(n)$ Laufzeit mit $O(n)$ Speicherplatz eine orthogonale Repräsentation berechnet werden, die das Restriktionensystem respektiert und für die es eine orthogonale planare Zeichnung im Kandinsky-Modell gibt.

BEWEIS Die Berechnung der orthogonalen Repräsentation erfolgt wie in Lemma 7 beschrieben. Außerdem soll der Graph durch die Datenstrukturen repräsentiert sein, die in Kapitel 5 zum Einfügen von Knoten und Kanten verwendet werden.

Jede Fläche speichert die Knoten eines Sektors in einem Sektorbaum, bei dem leicht überprüft werden kann, ob er leer ist bzw. nur aus einem einzigen Knoten besteht. Damit können in einem ersten Schritt die Bedingungen eins und zwei aus Lemma 7 überprüft werden und bei den gefundenen Kanten wird markiert, dass sie zwei Knicke haben. In der Summe kostet dies lineare Laufzeit, da es nur $O(n)$ Flächen mit je 4 Sektorbäumen gibt.

In einem zweiten Schritt werden die Kanten um jeden Knoten zyklisch im Uhrzeigersinn und dann gegen den Uhrzeigersinn durchlaufen. Während der Durchläufe können die Kanten markiert werden, die auf Grund der Regel drei oder vier zwei Knicke erhalten. Jede Kante wird vier mal besucht und da planare Graphen linear viele Kanten haben, erhalten wir wieder nur lineare Laufzeit.

Die Berechnung der Kantenverläufe aus diesen Informationen und den Attributen \preceq_x und \preceq_y kann für jede Kante dann in konstanter Zeit durchgeführt werden. In Summe erhalten wir lineare Laufzeit.

Im letzten Kapitel haben wir schon gezeigt, dass die bis dahin konstruierte Datenstruktur linearen Speicher benötigt. Hier benötigen wir ein zusätzliches Attribut zur Angabe, ob die Kante einen oder zwei Knicke hat. Außerdem müssen die Kantenverläufe gespeichert werden. Da die Kantenverläufe aus maximal drei Segmenten bestehen, können diese Informationen je Kante mit konstantem Speicherbedarf gespeichert werden und wir erhalten in Summe weiterhin linearen Speicherbedarf. ■

Der folgende Satz zeigt, dass mit der oben beschriebenen Konstruktion die maximale Anzahl an Kanten mit zwei Knicken der Anzahl der Flächen entspricht.

Satz 7

Gegeben sei ein Graph mit einem legalen Restriktionensystem. Dann kann daraus eine orthogonale Repräsentation konstruiert werden, die das Restriktionensystem respektiert und in der maximal $|F|$ Kanten zwei Knicke haben, wobei F die Menge der Flächen der Einbettung ist. Alle anderen Kanten haben einen Knick.

BEWEIS Die Berechnung der orthogonalen Repräsentation erfolgt wie in Lemma 7 beschrieben. Wir untersuchen deshalb die Regeln aus diesem Lemma, die festlegen, wann eine Kante zwei Knicke hat. Wir weisen jeder Kante, die auf Grund von Regel eins oder drei zwei Knicke hat, der Fläche oberhalb der Kante zu. Kanten, die wegen Regel zwei oder vier zwei Knicke haben, werden der Fläche unterhalb zugewiesen. Wir zeigen, dass jeder Fläche maximal eine Kante mit zwei Knicken zugewiesen ist.

Gegeben seien die Regeln eins und drei. Eine Kante e mit zwei Knicken wird einer Fläche f genau dann zugewiesen, wenn e in f die erste Kante in Sektor 4 ist. Da es in einem Sektor nur eine erste Kante geben kann, kann auf diese Art jeder Fläche nur eine Kante zugewiesen werden.

Ebenso kann mit den Regeln zwei und vier jeder Fläche nur eine Kante mit zwei Knicken zugewiesen werden, nämlich die letzte Kante in Sektor 1.

Schließlich müssen wir noch überprüfen, ob eine Kante wegen Regel eins oder drei und eine Kante wegen Regel zwei oder vier gemeinsam an die gleiche Fläche zugewiesen werden können. Sei dazu die Kante e_1 an Fläche f wegen Regel eins oder drei zugewiesen. In diesem Fall kann Sektor 2 in der Fläche f nicht leer sein.

Wenn der Fläche f nach Regel zwei oder vier eine Kante mit zwei Knicken zugewiesen wird, dann ist in f Sektor 2 leer. Deshalb kann keiner Fläche mehr als eine Kante mit zwei Knicken zugewiesen werden.

Insgesamt wird jeder Fläche also nur eine Kante mit zwei Knicken zugewiesen, woraus die Behauptung folgt. ■

Damit haben wir alle Ergebnisse und können sie auch auf ein statischer Szenario übertragen, um die Ergebnisse mit den Ergebnissen für statische Algorithmen zu vergleichen.

Satz 8

Gegeben sei ein planarer zweifach zusammenhängender Graph $G = (V, E)$.

Dann kann mit den Operationen *insertNode* und *insertEdge* in $O(n \log n)$ eine planare orthogonale Repräsentation des Graphen im Kandinsky-Modell erzeugt werden, in der die Summe der Kantenknicke kleiner oder gleich $|E| + |F|$ ist.

BEWEIS In Satz 5 und Satz 6 wird gezeigt, dass die Operationen *insertNode* und *insertEdge* mit logarithmischer Zeit implementiert werden können. In Summe kostet der Aufbau des Graphen also $O(n \log n)$. Das Zeichnen kann dann in Linearzeit erfolgen (siehe Lemma 8).

In Satz 7 wird gezeigt, dass nur maximal $|F|$ Kanten zwei Knicke haben. Daraus folgt, dass es in Summe $|E| + |F|$ Kanten gibt. ■

In [FK94] präsentieren Fößmeier und Kaufmann einen Algorithmus, der in $O(n^2 \log n)$ eine orthogonale Repräsentation eines beliebigen planaren Graphen im Kandinsky-Modell konstruiert. Die Anzahl an Kantenknicken ist optimal, wenn so genannte leere Flächen verboten sind. Jede Kante hat dabei maximal einen Knick.

Fößmeier, Kant und Kaufmann liefern in [FKK96] eine Linearzeitheuristik, die für einen gegebenen dreifach zusammenhängenden planaren Graphen eine orthogonale Repräsentation im Kandinsky-Modell konstruiert, in der jede Kante einen Knick hat.

In [BM97] zeigen Biedl und Kaufmann einen Algorithmus, der für einen gegebenen dreifach zusammenhängenden planaren Graphen in linearer Zeit eine orthogonale Repräsentation im Kandinsky-Modell konstruiert, in der jede Kante maximal einen Knick hat und in Summe gibt es $m - n$ Kantenknicke. Für beliebige planare Graphen führt das zu dem Ergebnis, dass in linearer Zeit eine orthogonale Repräsentation im Kandinsky-Modell konstruiert werden kann, in der jede Kante maximal einen Knick hat und in Summe gibt es maximal $3n - 8$ Kantenknicke ([Bie97]). Sie verbessern damit die Ergebnisse aus [FKK96].

Damit schneiden wir in der Summe der Kantenknicke durchweg schlechter ab. Gegenüber den dreifach zusammenhängenden Graphen sogar deutlich. Im Vergleich zu beliebigen planaren Graphen ergeben sich in Summe zirka $|F|$ zusätzliche Knicke.² Die

² Nach dem Satz von Euler gilt $m \leq 3n - 6$.

Linearzeitheuristiken erreichen ihre guten Ergebnisse dadurch, dass die Einfügereihenfolge nicht von einem Anwender vorgegeben ist, sondern sie verwenden die kanonische Ordnung. Deren Eigenschaften erlauben es, einzelne Kanten ohne Knick einzufügen. Die um $\log n$ erhöhte Laufzeit ergibt sich dadurch, dass interne Datenstrukturen aktuell gehalten werden müssen.

Bezüglich der Knickzahl schneidet [FKK96] am besten ab, was aber mit der höheren Laufzeit erkauft wird.

Wir schneiden also sowohl bei der Laufzeit als auch bei der Knickzahl schlechter ab. Allerdings haben wir dafür ein interaktives Zeichenverfahren.

Wir benötigen im Folgenden oft die Menge aller Kanten, die in der orthogonalen Repräsentation zwei Knicke haben. Deshalb vergeben wir den Namen E_2 dafür.

Definition 11 (E_2)

Gegeben sei ein Graph mit einem legalen Restriktionensystem. Wir bezeichnen die Menge aller Kanten, die in der zugehörigen orthogonalen Repräsentation nach Lemma 7 zwei Knicke haben, mit E_2 . ■

Bisher haben wir eine Obergrenze für die Anzahl an Kantenknicken angegeben. Diese Obergrenze kann auch (fast) erreicht werden. Die Abbildung 6.7 zeigt ein Beispiel, in dem $|F| - 1$ Kanten zwei Knicke haben.³

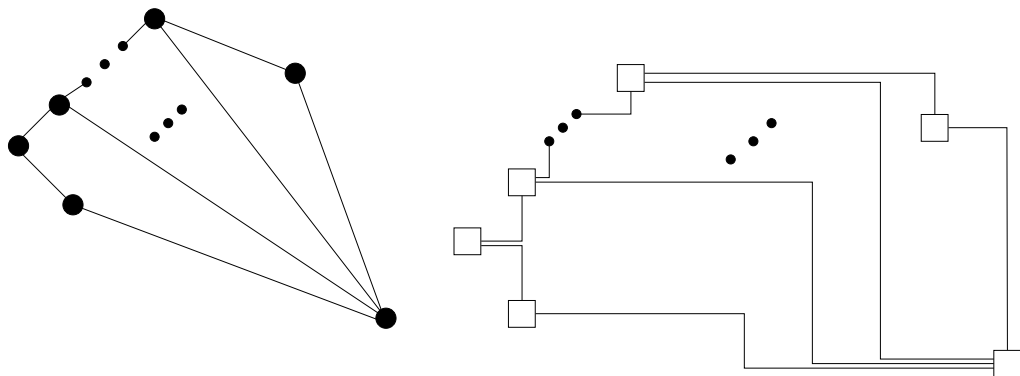


Abbildung 6.7: Worst-Case Konstruktion zur Anzahl an Kantenknicken.

Der hier beschriebene Algorithmus kann nur solange garantieren, dass sich ein Kantenverlauf nicht ändert, solange die Anzahl an Knicken für diese Kante sich nicht ändert. Dies kann aber passieren.

Diese Änderung der Kantenverläufe führt dazu, dass sich die Zeichnung des Graphen ändert. Wenn man aber mit relativ wenigen Kantenknicken auskommen will, dann muss man diesen Nachteil in Kauf nehmen. Auch diese Problematik gestaltet sich unserer Meinung nach aber in der Realität nicht so gravierend, wie es auf den ersten Blick

³ Man könnte hier durch „einfaches Hinsehen“ eine Zeichnung mit weniger Knicken finden. Das ist Thema des nächsten Kapitels.

scheinen mag. Das folgende Lemma zeigt, dass diese Änderungen des Kantenverlaufes in gewissem Sinn lokal sind.

Lemma 9 (Lokale Änderungen des Kantenverlaufes)

Gegeben sei ein Graph mit einem legalen Restriktionensystem und eine Kante, die bei der Generierung einer orthogonalen Repräsentation nach den Regeln in Lemma 7 nur einen Knick hat. Angenommen die Kante hat nach einer Einfügeoperation bei der Generierung einer orthogonalen Repräsentation zwei Knicke. Dann wurde bei einem der beiden Endknoten der Kante eine neue Kante eingefügt, oder bei einer der Kanten, die zu einem Endknoten adjazent ist, wurde ein Knoten eingefügt.

Weiter gilt: Gegeben sei eine Kante, die bei der Generierung einer orthogonalen Repräsentation nach den Regeln in Lemma 7 zwei Knicke hat. Angenommen, die Kante hat nach einer Einfügeoperation bei der Generierung einer orthogonalen Repräsentation nur einen Knick. Dann wurde bei einem der beiden Endknoten der Kante eine neue Kante eingefügt, oder bei einer der Kanten, die zu einem Endknoten adjazent ist, wurde ein Knoten eingefügt.

BEWEIS Regel drei und vier von Lemma 7 hängen nur von lokalen Eigenschaften am rechten Endknoten der Kante ab. Diese Eigenschaften können sich aber nur dann ändern, wenn an diesem Knoten eine Kante eingefügt wird, oder bei einer zu dem Knoten adjazenten Kante ein Knoten eingefügt wurde.

Regeln eins und zwei hängen von Flächeneigenschaften ab. Eine Kante mit zwei Knicken muss die einzige in einem Sektor der Fläche sein, nur der gegenüberliegende Sektor darf Knoten und Kanten enthalten. Auch diese Eigenschaft kann sich auf Grund der Regeln zur Neukonstruktion der orthogonalen Repräsentation nur dann ändern, wenn bei einem der Endknoten eine neue Kante eingefügt wird, oder bei einer zu einem Endknoten adjazenten Kante wurde ein Knoten eingefügt. ■

Das Verfahren in seiner bisherigen Form ist bezüglich der Knickzahl unbefriedigend. Wie man im obigen Beispiel sehen kann, hat die Zeichnung sehr viele Knicke. Schon durch einfaches „Hinschauen“ findet man viele Möglichkeiten zur Einsparung eines Knickes, falls \leq als Ordnungsrelation verwendet wird. Damit wollen wir uns im folgenden Kapitel beschäftigen.

7 Minimierung der Anzahl an Kantenknicken

Bisher haben wir einen Grundalgorithmus entwickelt, der im Wesentlichen aus zwei Komponenten besteht: Zum einen sind das die Funktionen zum interaktiven Aufbau eines Graphen und eines zugehörigen Restriktionensystems, das sicherstellt, dass die „mental map“ des Anwenders nicht zerstört wird. Die zweite Komponente ist ein Algorithmus, der aus dem Graph und dem Restriktionensystem eine orthogonale Repräsentation des Graphen konstruiert, für die dann mit bekannten Algorithmen eine Zeichnung generiert werden kann.

Der Zeichenalgorithmus konstruiert eine Zeichnung des Graphen, in der maximal $|F|$ Kanten zwei, die übrigen Kanten einen Knick haben. Da Kantenknicke ein wichtiges Qualitätskriterium beim orthogonalen Zeichnen von Graphen sind ([PCJ95]), beschäftigen wir uns in diesem Kapitel mit der Frage, auf welche Art die Kantenknicke reduziert werden können.

Dazu stellen wir zunächst Regeln auf, wie stark die Änderungen an den Zeichnungen sein dürfen. Danach untersuchen wir mit Worst-Case-Betrachtungen, wie weit die Anzahl an Kantenknicken überhaupt verringert werden kann, wenn die vorher definierten Rahmenbedingungen berücksichtigt werden müssen. Abschließend wird noch ein Algorithmus entwickelt, der im Rahmen der gesteckten Grenzen zu einer knickoptimale Zeichnung des Graphen führt. Das Ergebnis dieses Algorithmus ist dabei keine Zeichnung oder orthogonale Repräsentation, sondern das bisherige Restriktionensystem, angereichert um die Information, bei welchen Kanten sich welche Abweichungen vom bisherigen Kantenvverlauf ergeben. Der Algorithmus ergänzt also in einem Restriktionensystem zusätzliche Informationen, die beim Erzeugen der orthogonalen Repräsentation verwendet werden kann.

7.1 Rahmenbedingungen für Knickeinsparungen

Bisher interpretieren wir die Ordnungsrelation \preceq des Restriktionensystems als $<$ -Relation. Damit liegen zwei Knoten, die durch eine Kante verbunden sind, weder in der gleichen Spalte noch in der gleichen Zeile. Folglich hat jede Kante mindestens einen Knick.

In diesem Kapitel soll die Anzahl an Kantenknicken reduziert werden. Wie bisher auch, soll die Zeichnung nicht beliebig stark verändert werden. Andernfalls könnte man das

Ziel, eine Zeichnung mit weniger Knicken, mit einem beliebigen Optimierungsalgorithmus erreichen. Dabei würde aber keinerlei Rücksicht auf die „mental map“ des Anwenders genommen.

Wir müssen unser Zeichenverfahren also so anpassen, dass zum einen die „mental map“ des Anwenders möglichst gut erhalten bleibt und zum anderen in praktischen Anwendungen eine signifikante Reduktion der Anzahl an Kantenknicken möglich ist.

Wir betrachten die folgenden Anforderungen bei der Knickoptimierung, die im Anschluss formal definiert werden:

1. Die Einschränkungen zur Erhaltung der „mental map“ sind bei beiden Verfahren so zu gestalten, dass sich eine Zeichnung mit Knickoptimierung und eine Zeichnung, die ohne Knickoptimierung erzeugt wurde, ähneln. Dies führt dazu, dass selbst bei einem Wechsel zwischen zwei Zeichnungen, eine mit und eine ohne Knickoptimierung, die „mental map“ des Anwenders erhalten bleibt. Damit kann beliebig zwischen Knickminimierung und Standardalgorithmus gewechselt werden.
2. Aus der Literatur ([BT98]) ist bekannt, dass die Kantenverläufe ein wesentliches Element bei der „mental map“ darstellen. Kantenverläufe sollen also auch weiterhin so gut wie möglich erhalten bleiben.

Diese Überlegungen führen zu der folgenden Definition, in der das Restriktionensystem neu interpretiert wird.

Definition 12

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Eine orthogonale Zeichnung *respektiert das Restriktionensystem schwach*, wenn die folgende Eigenschaften erfüllt sind:

1. Die Zeichnung ist planar und die Einbettung der Zeichnung entspricht der Einbettung des Restriktionensystems.
2. Die Lagebeziehungen entsprechen den Attributen \preceq_x und \preceq_y der Kanten, wenn \preceq als \leq interpretiert wird.
3. Der Verlauf jeder einzelnen Kante entspricht dem Verlauf, den diese Kante hätte, wenn ihr Verlauf mit dem Algorithmus aus Kapitel 6 berechnet worden wäre. Der Verlauf einer Kante darf nur dann von dieser Regel abweichen, wenn dadurch bei der Kante selbst mindestens ein Kantenknick eingespart werden kann. ■

Diese Definition liefert also die Rahmenbedingung für die Knickoptimierung. Als Eingabe für den Zeichenalgorithmus erhalten wir weiterhin einen Graphen G mit dem schon bekannten Restriktionensystem. Allerdings interpretieren wir die Ordnungsrelation \preceq durch \leq anstelle von $<$. Die ersten beiden Bedingungen stellen also sicher, dass sich eine Zeichnung mit und eine Zeichnung ohne Knickoptimierung im Wesentlichen entsprechen. Die dritte Bedingung der Definition stabilisiert Kantenverläufe so weit möglich und erzwingt eine gewisse Lokalität der Optimierung. Mit dieser Bedingung ist es nicht möglich,

dass beliebig viele Kantenverläufe geändert werden, nur damit ein einziger Knick eingespart wird. Diese Lokalität ist bewusst gewählt, damit die „mental map“ des Anwenders erhalten bleibt.

Die Aufgabe in diesem Kapitel ist damit, die Menge der Kanten zu bestimmen, deren Kantenverlauf vom bisher berechneten Kantenverlauf abweicht. Dazu erweitern wir das Restriktionensystem um drei Mengen, in denen die Kanten gesammelt werden, die waagrecht, senkrecht, oder mit einem anstatt mit zwei Knicken gezeichnet werden. Klarerweise sind diese drei Mengen disjunkt, denn eine Kante kann z.B. nicht gleichzeitig waagrecht und senkrecht gezeichnet werden.

Definition 13 (Erweitertes Restriktionensystem)

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Zusätzlich seien drei Mengen E_w , E_s und E_1 gegeben. Dann wird $G = (V, E, R)$ mit E_w , E_s und E_1 als *erweitertes Restriktionensystem* bezeichnet.

Ein erweitertes Restriktionensystem heißt genau dann *legal*, wenn es dem 4-Sektorenmodell entspricht und eine orthogonale planare Zeichnung des Graphen im Kandinsky-Modell existiert, die das Restriktionensystem R schwach respektiert und zusätzlich folgende Eigenschaften gelten:

1. $\forall e \in E_w$: Die Kante e wird in der Zeichnung waagrecht ohne Knick gezeichnet.
2. $\forall e \in E_s$: Die Kante e wird in der Zeichnung senkrecht ohne Knick gezeichnet.
3. $\forall e \in E_1$: Die Kante e wird in der Zeichnung mit einem Knick gezeichnet, wobei die Kante abweichend von den bisherigen Anforderungen auf der Ober- oder Unterseite des linken Knotens beginnt und erst dann ein waagrechtes Kantenstück folgt. ■

Diese Definition liefert den Rahmen für dieses Kapitel: Als Eingabe ist ein Graph G mit einem Restriktionensystem R gegeben. Gesucht ist ein legales erweitertes Restriktionensystem, in dem die Summe der Kanten in den Mengen E_w , E_s und E_1 maximal ist, wobei Kanten, die ohne anstatt mit zwei Knicken gezeichnet werden, doppelt gezählt werden müssen.

Vor der Vorstellung eines Algorithmus zur Bestimmung dieser drei Mengen sollen noch die Grenzen der Knickreduktion abgesteckt werden.

7.2 Worst-Case Konstruktionen zur Knickeinsparung

In diesem Abschnitt werden wir anhand von Worst-Case Betrachtungen untersuchen, welche Ergebnisse wir bei der Knickoptimierung überhaupt erreichen können.

Wie schon mehrfach festgestellt, sind die Kantenverläufe ein wichtiges Element der „mental map“ und sollten nach Möglichkeit nicht geändert werden. Wenn wir hier die Anzahl an Kantenknicken reduzieren, in dem einzelne Kanten ohne Knick gezeichnet werden, dann stellt sich zunächst folgende Frage:

Kann sichergestellt werden, dass nach einer beliebigen Einfügeoperation alle Knickoptimierungen des Restriktionensystems auch im folgenden Restriktionensystem legal sind? Die Antwort liefert folgendes Lemma.

Lemma 10

Gegeben sei ein Graph G mit einem legalen erweiterten Restriktionensystem R mit den Mengen E_s, E_w und E_1 . Außerdem sei G' der Folgegraph einer restriktionenerhaltenden Einfügeoperation, die das Restriktionensystem R in ein Restriktionensystem R' überführt. Falls bei der Einfügeoperation eine der Kanten aus E_s, E_w oder E_1 ersetzt wird, dann wird die Kante mit der Einfügeoperation aus der Menge gelöscht.

Dann ist für den Graphen G' das erweiterte Restriktionensystem R' mit den Mengen E_s, E_w und E_1 im Allgemeinen nicht legal.

BEWEIS Wir zeigen das Lemma durch die Angabe eines Beispiels.

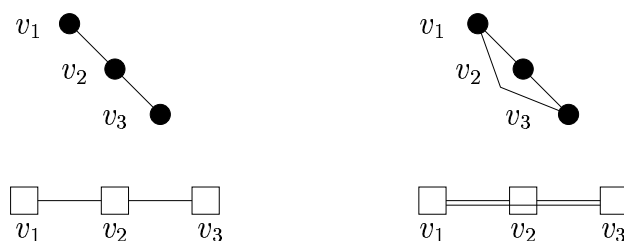


Abbildung 7.1: Eine neue Kante macht bisherige Einsparungen eines Knickes illegal.

Seien in Abbildung 7.1 die Kanten $\{v_1, v_2\}$ und $\{v_2, v_3\}$ Element von E_w . Gegeben sei außerdem eine restriktionenerhaltende Einfügeoperation, die eine neue Kante $\{v_1, v_3\}$ erzeugt.

Dann gibt es keine orthogonale planare Zeichnung des Graphen im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der die Kanten $\{v_1, v_2\}$ und $\{v_2, v_3\}$ waagrecht ohne Knick gezeichnet werden, denn die neue Kante $\{v_1, v_3\}$ müsste dann den Knoten v_2 schneiden. ■

Die Konsequenz aus diesem Lemma ist also, dass eine Menge von Optimierungen in einer Folgezeichnung nicht garantiert werden kann. Wir werden später noch sehen, dass die Menge an nicht mehr verwendbaren Optimierungen eingeschränkt werden kann.

Weiter wollen wir uns damit beschäftigen, ob bei einer Knickoptimierung eine untere Schranke für die Verbesserung bzgl. der Knickzahl gegeben werden kann.

Lemma 11

Es gibt beliebig große Graphen G mit legalem Restriktionensystem R , bei denen eine orthogonale planare Zeichnung im Kandinsky-Modell, die das Restriktionensystem schwach respektiert, in Summe nur 2 Knicke weniger haben kann als eine orthogonale planare Zeichnung im Kandinsky-Modell, die das Restriktionensystem respektiert.

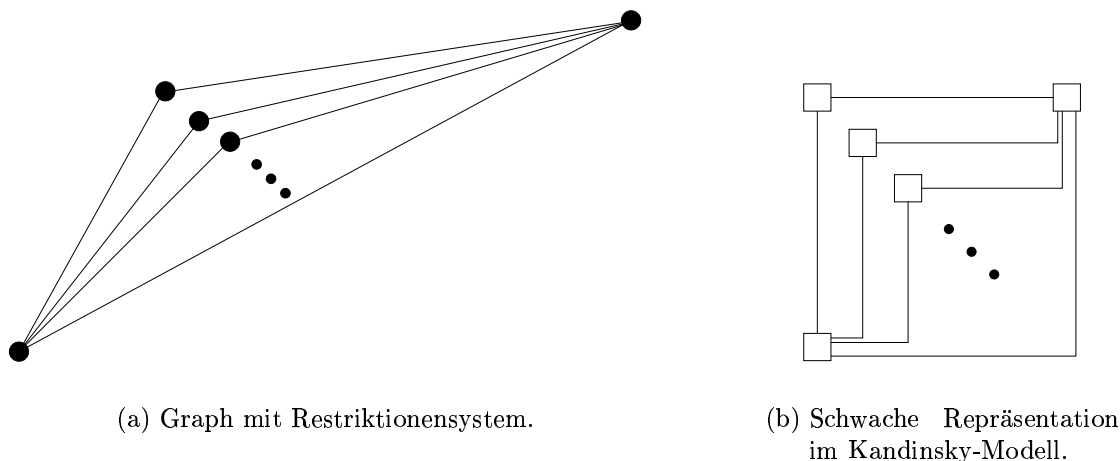


Abbildung 7.2: Graphfamilie mit Restriktionensystem, bei der nur zwei Knicke eingespart werden können.

BEWEIS Gegeben sei die Graphenfamilie mit Restriktionensystem aus Abbildung 7.2. Eine knickoptimale planare orthogonale Zeichnung des Graphen im Kandinsky-Modell hat in Summe $m-2$ Kantenknicke, wenn das Restriktionensystem schwach berücksichtigt werden muss. Denn alle Kanten innerhalb des umschließenden Rechteckes können nicht ohne Kantenknick gezeichnet werden, wenn das Restriktionensystem schwach respektiert werden muss. ■

Das Beispiel zeigt, dass es nicht möglich ist, deutlich weniger als m Kantenknicke zu erhalten.¹ Dann stellt sich aber noch die weitere Frage, ob wenigstens m Kantenknicke im Worst-Case genügen. Wir stellen uns also letztendlich die Frage, ob wenigstens bei den Kanten mit zwei Knicken eine Einsparung möglich ist. Die halbe Antwort auf diese Frage liefert wieder das folgende Lemma.

Lemma 12

Es gibt beliebig große Graphen G mit legalem Restriktionensystem R , bei denen eine orthogonale planare Zeichnung des Graphen im Kandinsky-Modell, die das Restriktionensystem schwach respektiert, in Summe mindestens $m + f/2 - 6$ Kantenknicke haben muss.

BEWEIS Betrachte dazu die Graphenfamilie mit Restriktionensystem aus Abbildung 7.3. Eine knickoptimale planare orthogonale Zeichnung des Graphen im Kandinsky-Modell hat in Summe $m + f/2 - 6$ Kantenknicke, wenn das Restriktionensystem schwach respektiert werden muss, das auf der rechten Seite dargestellt ist. Der äußere Rahmen erzwingt dabei die Kanten mit zwei Knicken. ■

¹ Auch ohne Berücksichtigung des Restriktionensystems können bei diesem Graph bei einer Zeichnung im Kandinsky-Modell nur eine konstante Anzahl an Kanten ohne Knick gezeichnet werden. Dies liegt daran, dass zwei Knoten hohen Grad haben; im Kandinsky-Modell können aber an einen Knoten immer nur maximal vier Kanten ohne Knick anschließen.

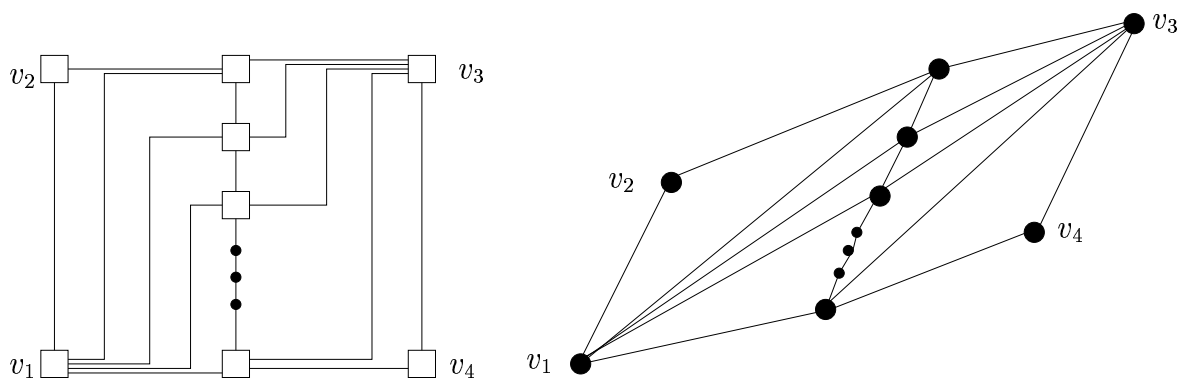


Abbildung 7.3: Nicht alle Kanten mit zwei Knicken können vermieden werden.

Bei der Konstruktion der orthogonalen Repräsentation des Graphen im letzten Lemma entsteht in der Mitte eine Sequenz von senkrechten Kanten ohne Knicke. Diese Sequenz sorgt dafür, dass man nicht $m + f - k$ Kantenknicke erhält. Solche Sequenzen haben wir bei allen Versuchen zu einer Worst-Case Konstruktion erhalten. Wir haben immer nur Möglichkeiten gefunden, um diese Sequenz von Kanten ohne Knicke zu verhindern, in dem wir je Kante der Sequenz eine neue Fläche ohne Kanten mit zwei Knicken oder wieder mit Kanten ohne Knick erstellt haben. Wir konnten aber auch nicht beweisen, dass man $f/2$ Kantenknicke einsparen kann.

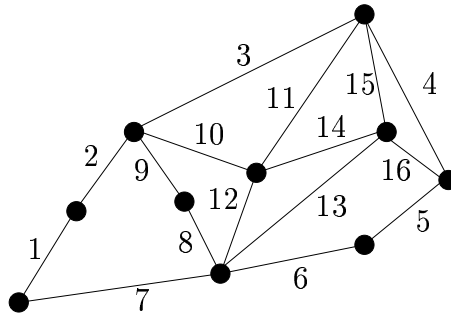
Diese Überlegungen liefern also in Summe ein recht ernüchterndes Ergebnis. Wir können im Worst-Case nur konstant viele Kantenknicke einsparen, sogar die Anzahl an Kanten mit zwei Knicken kann im Allgemeinen nicht reduziert werden. Zusätzlich können Optimierungen aus vorherigen Zeichnungen nicht garantiert übernommen werden².

Trotzdem macht eine Knickoptimierung für praktische Graphen Sinn. Die Konstruktionen, die dazu führen, dass eine Knickoptimierung nur sehr wenige Knicke einspart, sind sehr speziell. Wir wollen das mit dem kleinen, willkürlich gewählten Beispiel aus Abbildung 7.4 belegen. Es zeigt links das Restriktionensystem und rechts eine Zeichnung ohne Knickoptimierung und eine Zeichnung mit Knickoptimierung.

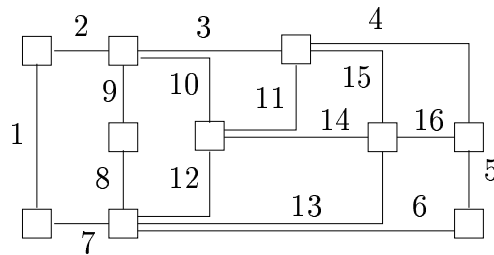
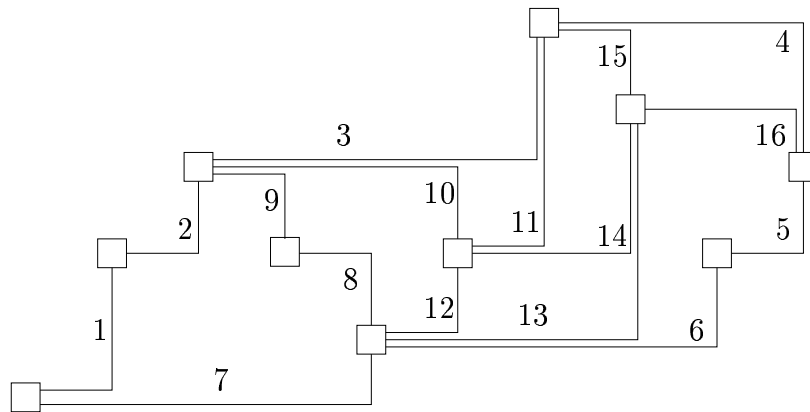
Nach diesen theoretischen Überlegungen wollen wir nun zeigen, wie eine Knickoptimierung innerhalb der gewählten Schranken erfolgen kann. Dabei gehen wir so vor, dass zunächst für jede Kante einzeln untersucht wird, ob diese Kante mit weniger Knicken gezeichnet werden kann. Das Ergebnis dieses Schrittes sind drei Mengen von Kanten, die senkrecht, waagrecht oder mit einem anstatt mit zwei Knicken gezeichnet werden können. Diese drei Mengen sind noch *nicht* identisch mit den drei Mengen E_w , E_s und E_1 aus Definition 13. Wir werden sehen, dass sich verschiedene Einsparungspotentiale gegenseitig ausschließen können.³ Es muss also aus diesen Mengen eine in Summe maximale Teilmenge gefunden werden, wobei alle Optimierungen aus diesen Mengen

² Wobei wir später sehen werden, dass dies nur in einer lokalen Umgebung der Einfügestelle gilt. Die Zeichnung ändert sich also nur an einer Stelle, an der auch der Graph geändert wird.

³ Es ist im Kandinsky-Modell z.B. nicht möglich, dass zwei Kanten den gleichen Knoten nach oben verlassen und beide Kanten senkrecht ohne Knick gezeichnet werden.



(a) Beispielgraph mit einem Restriktionensystem.



(b) Zeichnung mit und ohne Knickeinsparung.

Abbildung 7.4: Vergleich einer Zeichnung mit einer Knickeinsparung mit einer Zeichnung ohne Knickeinsparung.

verträglich sein müssen, also sich gegenseitig nicht ausschließen. Wir lösen diesen Schritt mit einer Transformation nach *maximum independent subset für bipartite Graphen* (vgl. [GJ83]). Die Transformation erfolgt, indem Optimierungsmöglichkeiten durch Knoten repräsentiert werden. Gegenseitige Ausschlüsse von Optimierungen führen zu einer Kante zwischen den Knoten, die diese Optimierungen repräsentieren.

Wir erhalten als Ergebnis nicht nur eine einfache Reduktion, sondern eine echte Optimierung der Knickzahl. Innerhalb der festgelegten Schranken gibt es damit keine Zeichnung mit weniger Knicken als das von uns berechnete Ergebnis.

Wir beginnen mit der Bestimmung der Mengen der möglichen Optimierungen. Die Abbildung 7.5 zeigt, wie Knicke eingespart werden können.

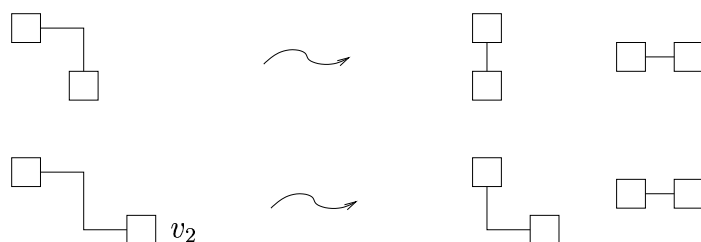


Abbildung 7.5: Optimierungsmöglichkeiten durch eine Änderung der Ordnungsrelation.

7.3 Knickeinsparungen bei einzelnen Kanten

In diesem Abschnitt beschränken wir uns zunächst nur auf eine einzige Kante. Wir untersuchen die folgende Frage:

Gegeben sei ein Graph $G = (V, E)$ mit einem Restriktionensystem R und eine beliebige Kante $e \in E$. Gibt es eine orthogonale planare Zeichnung des Graphen im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der die Kante e waage- oder senkrecht ohne Knick gezeichnet wird?

Wir gehen damit in diesem Abschnitt *nicht* auf die Frage ein, welche verschiedenen Optimierungsmöglichkeiten sich gegenseitig ausschließen. Wir stellen für jede Kante einzeln fest, ob auf dieser Kante ein Knick eingespart werden kann, ohne den Verlauf der übrigen Kanten zu ändern. Gegenseitige Wechselwirkungen von Einsparungen eines Knickes werden ignoriert.

Wir beginnen mit einigen grundsätzlichen Überlegungen, die bei der Notation der folgenden Beweise von Nutzen sind.

Bedingt durch die „bend or end“ Eigenschaft im Kandinsky-Modell können nicht beliebige Kanten ohne Knick auf einer Seite eines Knotens enden. Betrachten wir dazu alle Kanten, die auf derselben Seite eines Knotens enden. Eine Kante e aus dieser Menge kann nur dann ohne Knick gezeichnet werden, wenn die, in der zyklischen Ordnung der Kanten um den Knoten vor e kommenden Kanten, nach links abknicken und die Nachfolger nach rechts.

Durch das Restriktionensystem ist bei uns festgelegt, welche Kanten einen Linksknick und welche einen Rechtsknick haben müssen. Das ergibt sich aus der Tatsache, dass die relative Lage von adjazenten Knoten fixiert ist. Damit kann es für jede Seite eines Knotens immer nur zwei Kanten geben, die potentiell auf dieser Seite des Knotens enden und keinen Kantenknick haben. Das sind die in der zyklischen Reihenfolge der Kanten die letzte, die nach links knickt und die erste, die nach rechts knickt. Nur diese beiden Kanten je Seite eines Knotens sind für die aktuellen Betrachtungen relevant. Um sie leichter identifizieren zu können, führen wir zusätzliche Objekte ein, die wir *Senkrechten* nennen. Sie werden in der Liste der Kanten um einen Knoten so platziert, dass eine Kante nur dann ohne Knick auf einer Seite eines Knotens enden kann, wenn sie direkter Nachfolger oder Vorgänger der Senkrechten der Seite ist.

Definition 14 (Senkrechten)

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $v \in V$ und die zyklische Liste L der zu v adjazenten Kanten gegeben.

Ein neu in die Liste L der zu v adjazenten Kanten eingefügtes Objekt heißt

1. *Senkrechte links* und kurz $s_l \Leftrightarrow$ Der zyklische Vorgänger verlässt den Knoten nach unten, der zyklische Nachfolger nach oben.
2. *Senkrechte oben* und kurz $s_o \Leftrightarrow$ Der zyklische Vorgänger verlässt den Knoten nach links, der zyklische Nachfolger nach rechts.
3. *Senkrechte rechts* und kurz $s_r \Leftrightarrow$ Der zyklische Vorgänger verlässt den Knoten nach oben, der zyklische Nachfolger nach unten.
4. *Senkrechte unten* und kurz $s_u \Leftrightarrow$ Der zyklische Vorgänger verlässt den Knoten nach rechts, der zyklische Nachfolger nach links. ■

Abbildung 7.6 zeigt einen Knoten mit den adjazenten Kanten und Senkrechten. Die Richtung der Kanten ist jeweils so gezeichnet, wie sich das aus den Lagebeziehungen des Restriktionensystems ergibt.

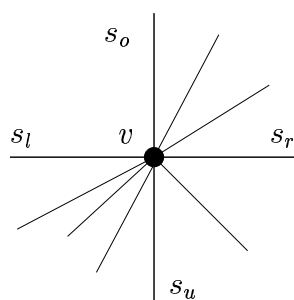


Abbildung 7.6: Adjazente Kanten und Senkrechten des Knotens v .

Diese zusätzlichen Objekte können in die Liste der Kanten um die Knoten eines Graphen in Linearzeit eingefügt werden, was im Folgenden Lemma gezeigt wird.

Lemma 13

Gegeben sei ein Graph $G = (V, E)$ mit einem Restriktionensystem R . Die Senkrechten s_o, s_r, s_u, s_l können für alle $v \in V$ in Summe in $O(n)$ eingefügt werden.

BEWEIS Zur Berechnung genügt ein Lauf in zyklischer Reihenfolge durch alle Kanten, die zu einem Knoten adjazent sind. Während des Durchlaufes werden die Attribute \preceq_x und \preceq_y untersucht und damit die Richtung festgestellt, in die die Kante den Knoten verlässt. Damit wird jede Kante zweimal besucht und, da die Anzahl an Kanten in einem planaren Graphen linear ist, erhalten wir $O(n)$ Laufzeit. ■

Bei der Reduzierung der Kantenknick gehen wir mit der gleichen Idee vor, wie in [Fös97] bei Zeichnungen mit dem Treppenalgorithmus die Knickzahl in einem zweiten Schritt reduziert wird. Die Idee wird im Folgenden kurz erläutert.

Wir gehen davon aus, dass bereits eine orthogonale Repräsentation des Graphen mit dem Algorithmus aus Kapitel 5 berechnet wurde. Wenn eine Kante davon abweichend jetzt ohne Knick gezeichnet werden soll, dann müssen die beiden Endknoten in eine gemeinsame Zeile oder Spalte verschoben werden. Es gibt Bedingungen, die erfüllt sein müssen, damit die Endknoten einer Kante in einer gemeinsamen Zeile oder Spalte liegen können. Diese Bedingungen wollen wir in Anlehnung an [Fös97] *Verschiebungsbedingungen* nennen. Die Aufgabe in diesem Abschnitt ist es, die Verschiebungsbedingungen zu finden. Außerdem müssen wir zeigen, wie für eine Kante diese Bedingungen überprüft werden können.

Die Voraussetzungen leiten sich aus dem verwendeten Zeichenmodell ab. Wenn eine Kante ohne Knick gezeichnet wird, dann dürfen dadurch die Anforderungen des Kandinsky-Modells nicht verletzt werden. Daraus können Eigenschaften eines Restriktionensystems abgeleitet werden, die erfüllt sein müssen, damit eine Kante ohne Knick gezeichnet werden kann. Wir suchen also Eigenschaften des Restriktionensystems, die die Verschiebungsbedingungen repräsentieren. Wir erlauben im Gegensatz zum Kandinsky-Algorithmus so genannte leere Flächen ([FK95]), wenn dadurch die Planarität nicht zerstört wird.

Als Grundlage zeigen die folgenden Lemmas notwendige⁴ Eigenschaften eines Restriktionensystems, die erfüllt sein müssen, damit eine Kante ohne Knick gezeichnet werden kann.

Lemma 14

Gegeben seien ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E$ eine beliebige Kante aus der Kantenmenge. O.B.d.A. sei $\{v_1, v_2\} \cdot \preceq_x = \text{true}$.⁵

Es kann nur dann eine orthogonale planare Zeichnung des Graphen im Kandinsky-Modell existieren, die das Restriktionensystem schwach respektiert und in der e waagrecht ohne Knick gezeichnet wird, wenn die Kante e am Knoten v_1 direkter Nachbar von s_r und am Knoten v_2 direkter Nachbar von s_l ist.

⁴ aber nicht hinreichende

⁵ Das heißt, v_1 liegt links von v_2 .

BEWEIS Nehmen wir an, dass e am Knoten v_1 nicht direkter Nachbar von s_r ist. Wegen der Eigenschaften des Kandinsky-Modells kann nur eine Kante auf einer Seite eines Knotens enden und ohne Knick gezeichnet werden. Zwischen e und s_r gibt es also eine Kante, die einen Knick haben muss und auf Grund der Lagebeziehungen aus dem Restriktionensystem muss diese Kante entweder e oder v_2 schneiden und die Zeichnung kann nicht planar sein (vgl. Abbildung 7.7).

Eine entsprechende Argumentation zeigt die Behauptung für v_2 und die Senkrechte s_l . ■

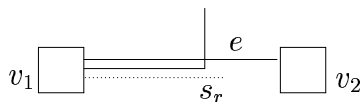


Abbildung 7.7: Illustration zu Lemma 14, das zeigt, dass nur ein direkter Nachbar einer Senkrechte in einer planaren Zeichnung ohne Knick gezeichnet werden kann.

Lemma 15

Gegeben seien ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E$ eine beliebige Kante aus der Kantenmenge. O.B.d.A. sei $\{v_1, v_2\}. \preceq_y = false$.⁶

Es kann nur dann eine orthogonale planare Zeichnung des Graphen im Kandinsky-Modell existieren, die das Restriktionensystem schwach respektiert und in der e senkrecht ohne Knick gezeichnet wird, wenn die Kante e am Knoten v_1 direkter Nachbar von s_u und am Knoten v_2 direkter Nachbar von s_o ist.

BEWEIS Der Beweis kann analog zu dem Beweis aus Lemma 14 geführt werden. ■

Das folgende Lemma beschäftigt sich mit der Eigenschaft aus dem Kandinsky-Modell, dass bei einer Kante ohne Kantenknick der zweite Endknoten immer „hinter“ den Kantenknicken aller anderen Kanten liegen muss, die auf der gleichen Seite des Knotens enden. In Abbildung 7.8 ist das graphisch dargestellt. Da auch in der Literatur zum Kandinsky-Modell dafür kein spezieller Begriff geprägt wird, verwenden wir ebenfalls die umgangssprachlichen Beschreibungen, dass die Kantenknicke „vor“ dem Endknoten der Kante ohne Knicke liegen, bzw. der Endknoten „hinter“ den Kantenknicken liegt.

Lemma 16

Gegeben seien ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E$ eine beliebige Kante aus der Kantenmenge.

Eine orthogonale planare Zeichnung Z des Graphen im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der e waage- oder senkrecht ohne Knick gezeichnet wird, existieren nur dann, wenn keine Kante e' existiert, die in Z gemeinsam mit e auf der gleichen Seite eines der beiden Endknoten v_1 oder v_2 endet und auf

⁶ Das heißt, v_1 liegt oberhalb von v_2 .

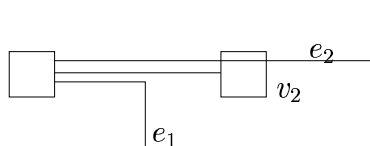


Abbildung 7.8: Kante e_2 liegt „hinter“ v_2 und verursacht damit einen Knoten-Kanten-Schnitt. Dagegen erfüllt e_1 die in Lemma 16 geforderte Eigenschaft.

Grund der Lagebeziehungen des Restriktionensystems muss der Kantenknick „hinter“ dem Knoten folgen.

BEWEIS Direkte Konsequenz aus den Eigenschaften des Kandinsky-Modells. ■

Das Restriktionensystem legt eine Menge von Lagebeziehungen fest. Daneben gibt es noch zusätzliche Einschränkungen, ohne deren Berücksichtigung keine planare Zeichnung möglich ist. In Abbildung 7.9 ist ein Beispiel dafür dargestellt. Hier muss Knoten v_3 in einer planaren Zeichnung unterhalb von Knoten v_2 liegen, obwohl dies durch das Restriktionensystem nicht direkt gefordert wird. Wenn unsere Anforderungen für eine Zeichnung erfüllt sein müssen und v_2 unterhalb von v_3 liegen würde, dann hätte die Kante $\{v_3, v\}$ einen Knick und würde die Kante $\{v_1, v_2\}$ schneiden. Auch solche indirekten Lagebeziehungen müssen bei einer Knickoptimierung berücksichtigt werden. So wird hier z.B. verhindert, dass die Kante $\{v_1, v_2\}$ senkrecht ohne Knick gezeichnet wird, was möglich wäre, wenn v_2 in einer orthogonalen planaren Zeichnung, die das Restriktionensystem schwach repräsentiert, unterhalb von v_3 liegen könnte. Damit in diesem Beispiel also die Kante $\{v_1, v_2\}$ senkrecht ohne Knick gezeichnet werden könnte, müsste es eine planare Zeichnung geben, in der v_3 oberhalb von v_2 liegt.

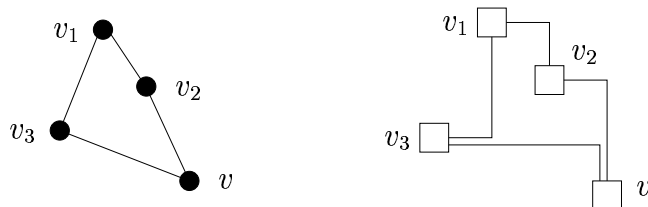


Abbildung 7.9: Der Knoten v_3 muss unterhalb v_2 liegen, obwohl dies durch das Restriktionensystem nicht erzwungen wird.

Wir geben nun die Bedingungen an, wann eine Kante senkrecht oder waagrecht ohne Knick gezeichnet werden kann. Sie ergeben sich direkt aus allen obigen Anforderungen.

Lemma 17

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E$ eine beliebige Kante aus der Kantenmenge und o.B.d.A. sei $\{v_1, v_2\} \preceq_y = \text{false}$ ⁷.

⁷ Das heißt, v_1 liegt in einer Zeichnung oberhalb von v_2 .

Eine orthogonale planare Zeichnung Z des Graphen im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der e senkrecht ohne Knick gezeichnet wird, existiert genau dann, wenn folgende Bedingungen gelten:

1. *Es gibt neben der Kante e keine Sequenz von Kanten, deren Lageabhängigkeiten v_1 und v_2 in x -Richtung bzgl. \preceq ordnen, so dass auf Grund dieser Sequenz $v_1 \preceq_x v_2$ gilt.*
2. *Die Kante e ist am Knoten v_1 direkter Nachbar von s_u und am Knoten v_2 direkter Nachbar von s_o .*
3. *Sei $e' = \{v_1, v_3\}$ neben e der zweite Nachbar der Senkrechten s_u am Knoten v_1 . Wenn e' wie e in Z auf der Unterseite von v_1 endet, dann gibt es im Restriktionensystem keine Lageabhängigkeiten, so dass (auch transitiv) v_3 unterhalb von v_2 liegen muss.⁸*
4. *Sei $e'' = \{v_2, v_4\}$ neben e der zweite Nachbar der Senkrechten s_o am Knoten v_2 . Wenn e'' wie e in Z auf der Oberseite von v_2 endet, dann gibt es im Restriktionensystem keine Lageabhängigkeiten, so dass (auch transitiv) v_4 oberhalb von v_1 liegen muss.*
5. *Sei $e' = \{v_1, v_3\}$ neben e der zweite Nachbar der Senkrechten s_u am Knoten v_1 . Wenn e' wie e in Z auf der Unterseite von v_1 endet, dann gibt es eine planare Zeichnung des Graphen, in der v_3 oberhalb von v_2 liegt.⁹*
6. *Sei $e'' = \{v_2, v_4\}$ neben e der zweite Nachbar der Senkrechten s_o am Knoten v_2 . Wenn e'' wie e in Z auf der Oberseite von v_2 endet, dann gibt es eine planare Zeichnung des Graphen, in der v_4 unterhalb von v_1 liegt.*

BEWEIS

\Rightarrow : Die geforderten Bedingungen 2 bis 6 sind eine direkte Konsequenz aus den Lemmas 14, 15 und 16 bzw. aus der „bend-or-end“-Eigenschaft im Kandinsky-Modell.

Wenn Bedingung 1 nicht erfüllt ist, dann gibt es einen Zyklus aus Kante e und der Sequenz von Knoten zwischen v_1 und v_2 . Der Zyklus müsste vollständig in einer Spalte liegen, was aber bei einer planaren Zeichnung nicht möglich ist.

\Leftarrow : Es ist zu zeigen, dass eine planare, orthogonale Zeichnung des Graphen existiert, in der e senkrecht ohne Knick gezeichnet werden kann.

Wir wissen schon, dass es eine orthogonale planare Zeichnung gibt, die das Restriktionensystem respektiert, wenn der Kantenverlauf mit dem Algorithmus aus Kapitel 6 berechnet wird. Die beiden Endknoten der Kante v_1 und v_2 definieren ein Rechteck, wenn man die beiden Knoten als zwei gegenüberliegende Eckpunkte interpretiert. Gemäß der Idee der Verschiebebedingungen müssen die angegebenen Bedingungen sicherstellen, dass

⁸ Dies würde zu einer Situation führen, die ähnlich zur Abbildung 7.8 ist.

⁹ Die Situation ist in Abbildung 7.9 dargestellt.

die Knoten v_1 und v_2 in eine gemeinsame Spalte verschoben werden können, o.B.d.A v_2 in die Spalte von v_1 .

Da e Nachbar der beiden Senkrechten ist, gibt es keine zu v_1 oder v_2 adjazente Kante, die die Kante e schneiden muss, wenn e senkrecht ohne Knick gezeichnet wird, denn alle Kanten links von e knicken zur linken Seite ab und alle Kanten rechts zur rechten Seite.

Die Eigenschaft aus Lemma 16, dass ein Kantenknicke vor dem Endknoten einer Kante ohne Knick liegen muss, ist ebenfalls erfüllt:

Alle Kanten, die in einer orthogonalen Repräsentation, die mit dem Algorithmus aus Kapitel 6 erstellt wurde, einen der Endknoten auf der gleichen Seite wie e verlassen und zur gleichen Seite abknicken, können vor dem zweiten Endknoten abknicken, sonst gäbe es auch eine Verletzung der Planarität, wenn man e mit einem Knick zeichnen würde. Es fehlen also noch die Kanten, die zwar auf der gleichen Seite eines der Endknoten anschließen, aber nach der anderen Seite abknicken. Die Bedingungen drei bis sechs stellen aber sicher, dass der Endknoten der zweiten Kante, die an einem der Endknoten Nachbar der Senkrechten ist, in einer Spalte näher zum Endknoten liegen kann als der zweite Endknoten der Kante e und damit liegt der Knick dieser Kante vor dem Endknoten der Kante ohne Knick. Alle anderen Kanten, die zu dieser Seite abknicken, können schon vor diesem Knick abknicken, sonst gäbe es schon für die orthogonale Repräsentation aus Kapitel 6 keine planare orthogonale Zeichnung. ■

Ein ähnliches Lemma erhalten wir bei der Untersuchung, welche Kanten waagerecht ohne Knick gezeichnet werden können.

Zum Beweis von diesem benötigen wir aber noch ein anderes Lemma.

Lemma 18

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Seien $e_1 = \{v_1, v\}$ und $e_2 = \{v_2, v\}$ zwei Kanten, die beide Nachbarn der Senkrechte s_l von Knoten v sind.

Wenn e_1 bei der Berechnung einer orthogonalen Repräsentation in Kapitel 6 zwei Knicke hat, dann gibt es eine orthogonale planare Zeichnung des Graphen, die das Restriktionensystem respektiert und in der die Spalte des senkrechten Kantenstückes von e_1 rechts der Spalte von v_2 liegt.

BEWEIS Wenn der senkrechte Teil von e_1 links von v_2 liegen müsste, dann muss es nach dem Lemma 6 in der Fläche zwischen den beiden Kanten e_1 und e_2 eine Sequenz von Kanten geben, so dass das senkrechte Kantenstück bzgl. \preceq in x-Richtung entsprechend angeordnet ist. Dazu müssten ein senkrecht Kantenstück und ein Knoten geordnet sein. Da es aber keine direkte Lageabhängigkeit zu einem senkrechten Kantenstück gibt, kann es nur eine Lageabhängigkeit zwischen dem Endknoten v_1 und dem Knoten v_2 geben. Eine solche Lageabhängigkeit kann aber nur erzwingen, dass das senkrechte Kantenstück rechts von v_2 liegen muss und nicht links. ■

Lemma 19

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E$ eine beliebige Kante aus der Kantenmenge und o.B.d.A. sei $\{v_1, v_2\} \cdot \preceq_x = \text{true}$ ¹⁰.

Es gibt genau dann eine orthogonale planare Zeichnung Z des Graphen im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der e waagerecht ohne Knick gezeichnet wird, wenn

1. Es gibt neben der Kante e keine Sequenz von Kanten, deren Lageabhängigkeiten v_1 und v_2 in y -Richtung bzgl \preceq ordnen, so dass auf Grund dieser Sequenz $v_1 \preceq_y v_2$ gelten muss.
2. Die Kante e ist am Knoten v_1 direkter Nachbar von s_r und am Knoten v_2 direkter Nachbar von s_l .
3. Sei $e' = \{v_1, v_3\}$ neben e der zweite Nachbar der Senkrechten s_r am Knoten v_1 . Wenn e' wie e in Z auf der rechten Seite von v_1 endet, dann gibt es im Restriktionensystem keine Lageabhängigkeiten, so dass (auch transitiv) v_3 rechts von v_2 liegen muss, außer e' hat zwei Knicke.
4. Sei wieder $e' = \{v_1, v_3\}$ neben e der zweite Nachbar der Senkrechten s_r am Knoten v_1 . Wenn e' nur einen Knick hat und wie e in Z auf der rechten Seite von v_1 endet, dann gibt es eine planare Zeichnung des Graphen, in der v_3 links von v_2 liegt.

BEWEIS

\Rightarrow : Die geforderten Bedingungen 2 bis 4 sind eine direkte Konsequenz aus den Lemmas 14, 15 und 16 bzw. aus der „bend-or-end“-Eigenschaft im Kandinsky-Modell.

Wenn Bedingung 1 nicht erfüllt ist, dann gibt es einen Zyklus aus Kante e und der Sequenz von Knoten zwischen v_1 und v_2 . Der Zyklus müsste vollständig in einer Zeile liegen, was aber bei einer planaren Zeichnung nicht möglich ist.

\Leftarrow : Wir gehen wieder wie in Lemma 17 vor und zeigen, dass v_2 in die Zeile von v_1 verschoben werden kann.

Da e Nachbar der beiden Senkrechten ist, gibt es keine zu v_1 oder v_2 adjazente Kante, die die Kante e schneiden muss, wenn e waagerecht ohne Knick gezeichnet wird, denn alle Kanten links von e knicken zur linken Seite ab und alle Kanten rechts zur rechten Seite.

Die Eigenschaft aus Lemma 16, dass ein Kantenknick vor dem Endknoten einer Kante ohne Knick liegen muss, ist ebenfalls erfüllt, denn es gilt für den Knoten v_1 die gleiche Argumentation wie in Lemma 17. Bei Knoten v_2 können auf Grund der Konstruktion der orthogonalen Repräsentation nur Kanten auf der linken Seite des Knotens enden, wenn die Kante zwei Knicke hat. Der senkrechte Teil einer Kante mit zwei Knicken kann aber immer näher am rechten Endknoten liegen als der Endknoten einer Kante mit einem oder keinem Knick (siehe Lemma 18). ■

¹⁰ Das heißt, v_1 liegt in einer Zeichnung links von v_2 .

Damit haben wir die Bedingungen, wann eine einzelne Kante waage- oder senkrecht ohne Knick gezeichnet werden kann. Es bleibt zu zeigen, wie und mit welcher Komplexität diese Bedingungen aus einem Graph und einem Restriktionensystem überprüft werden können.

Das folgende Lemma zeigt Eigenschaften einzelner Attribute eines Restriktionensystems, die erfüllt sein müssen, damit eine Kante senkrecht gezeichnet werden kann. Wir zeigen die Eigenschaften nur für den Verlauf einer Kante, deren Lageabhängigkeiten festlegen, dass der linke Knoten immer oberhalb des rechten Endknotens liegt. Da Zeichnungen im 4-Sektorenmodell achsensymmetrisch bzgl. der x-Achse sind, kann der andere Fall direkt daraus abgeleitet werden.

Lemma 20

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E$ eine beliebige Kante aus der Kantenmenge und o.B.d.A. seien $\{v_1, v_2\}. \preceq_x$ und $\{v_1, v_2\}. \preceq_y$ so belegt, dass $d(e, v_1) = \searrow$.

Es gibt genau dann eine orthogonale planare Zeichnung Z des Graphen im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der e senkrecht ohne Knick gezeichnet wird, wenn

1. $e.succ(v_1) = s_u \wedge e.succ(v_2) = s_o$
2. Sei $\{v_1, v_3\} = s_u.succ(v_1)$ die nächste Kante nach der Senkrechte an der Unterseite von v_1 . Falls $d(\{v_1, v_3\}, v_1) = \swarrow$, dann muss gelten: Es existieren im Restriktionensystem keine Lageabhängigkeiten, so dass v_3 unterhalb von v_2 liegen muss.
3. Sei $e' = e.succ(v_1)$. Wenn $d(e', v_1) = \swarrow$ gilt, Dann dürfen in der Fläche f , die unterhalb von e und e' liegt, nicht gleichzeitig
 - a) Sektor 3 leer sein,
 - b) Sektor 1 und 4 nur aus einer Kante bestehen und
 - c) in Sektor 2 zusätzlich zu Kante e noch mindestens eine weitere Kante liegen.¹¹

BEWEIS Wir zeigen die Äquivalenz der in diesem Lemma geforderten Eigenschaften und der Eigenschaften aus Lemma 17.

Bedingung 1 stellt sicher, dass die Kante e direkter Nachbar der Senkrechte s_u bei Knoten v_1 und der Senkrechte s_o bei Knoten v_2 ist. Die Eigenschaft ist damit zur Eigenschaft 2 aus Lemma 17 äquivalent. Außerdem ist Bedingung 2 äquivalent zu Bedingung 3 in Lemma 17. Zusätzlich gilt, dass der zweite Nachbar der Senkrechte s_o von v_2 nicht auf der Oberseite von v_2 enden kann. Denn jede Kante mit mindestens einem Knick beginnt auf der rechten Seite des linken Endknotens. Damit sind die Voraussetzungen für die Eigenschaften 4 und 6 in Lemma 17 nie gegeben. Wir müssen die Äquivalenz also nur noch für die übrigen Eigenschaften 1 und 5 aus Lemma 17 und die Bedingung 3 aus diesem Lemma zeigen.

¹¹ Abbildung 7.10 zeigt eine Illustration, bei der diese Bedingungen gleichzeitig gelten.

\Rightarrow : (Die hier geforderten Eigenschaften implizieren die Eigenschaften aus Lemma 17.)

Sei $e' = \{v_1, v_3\}$ die Kante aus E , die auf der Unterseite von v_1 endet und für die $d(e', v_1) = \swarrow$ gilt. Wenn es eine solche Kante gibt, dann muss für diese Kante die Eigenschaft 5 in Lemma 17 gelten. Gebe es also eine solche Kante. Für Bedingung 5 in Lemma 17 muss es eine planare Zeichnung geben, in der v_3 oberhalb von v_2 liegt. Dabei wissen wir aus Bedingung 2, dass es keine Lageabhängigkeiten gibt, die implizieren, dass v_3 unterhalb von v_2 liegen muss. Sei f die Fläche, die die Kanten e und e' gemeinsam hat und unterhalb dieser Kanten liegt.

Es gibt keine planare Zeichnung, in der v_3 oberhalb von v_2 liegt, wenn auf der rechten Seite von v_3 eine Kante beginnt, deren rechter Endknoten rechts von v_2 liegen muss, denn diese Kante würde dann die Kante e schneiden, wenn Kante e senkrecht ohne Knick gezeichnet wird. Das ist aber nur dann der Fall, wenn der zweite Endknoten v_4 dieser Kante nicht identisch mit v_2 ist und außerdem eine Lageabhängigkeit zwischen v_2 und v_4 besteht, so dass v_4 rechts von v_2 liegen muss. Wegen Lemma 6 muss der Pfad, der die Lageabhängigkeit impliziert, dann aber auch in Fläche f vorhanden sein. Die einzige Möglichkeit dazu wird aber durch Bedingung 3 aus diesem Lemma verboten.

Damit muss noch gezeigt werden, dass Bedingung 1 aus Lemma 17 erfüllt ist. Es darf also neben der Kante e keine weitere Sequenz von Kanten geben, die v_1 und v_2 in x -Richtung bzgl. \preceq ordnen. Angenommen, es gibt so eine Sequenz. Dann müsste die erste Kante Knoten v_1 nach rechts verlassen und Knoten v_2 nach links und der Pfad muss vollständig ober- oder unterhalb e liegen. Dann kann aber e nicht wie in Bedingung 1 gefordert zu den beiden Senkrechten benachbart sein und die Annahme ist falsch.

\Leftarrow : (Die Eigenschaften aus Lemma 17 implizieren die hier geforderten Eigenschaften.)

Wenn die Fläche f so aufgebaut wäre, wie es durch Bedingung 3 gefordert wird, dann gäbe es nur eine planare Zeichnung, wenn v_3 unterhalb von v_2 liegt. Nach Bedingung 5 aus Lemma 17 ist das aber verboten. ■

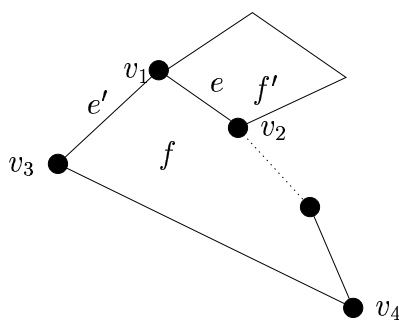


Abbildung 7.10: Bezeichnungen zu Lemma 20.

Das folgende Lemma zeigt, mit welcher Komplexität die hier geforderten Eigenschaften überprüft werden können.

Lemma 21

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Dann kann für jede Kante in $O(1)$ bestimmt werden, ob sie die Verschiebeeigenschaften für eine senkrechte Kante ohne Knick erfüllt.

BEWEIS Es sind die Eigenschaften aus Lemma 20 zu prüfen.

Die Bedingungen eins und drei können direkt aus den Kantenattributen berechnet werden, indem wir die Richtungen der betrachteten Kanten untersuchen.

Für Bedingung zwei muss festgestellt werden, ob auf Grund des Restriktionensystems eine Lageabhängigkeit zwischen den Knoten v_2 und v_3 existiert, so dass v_3 unterhalb von v_2 liegen muss. Nach Lemma 6 existiert nur dann eine Lageabhängigkeit, wenn diese auch auf Grund von Kanten innerhalb einer gemeinsamen Fläche existiert. Dazu betrachten wir die Fläche, die e und e' gemeinsam haben. Eine Abhängigkeit existiert genau dann, wenn in f der Knoten v_3 den Anfang von Sektor 1 markiert und Sektor 4 leer ist. Dies kann in $O(1)$ aus den Attributen \preceq_x und \preceq_y der Kante $v_3.succ(e')$ berechnet werden. ■

Korollar 1

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Dann kann in $O(n)$ die Menge E_s aller Kanten bestimmt werden, die in einer orthogonalen planaren Zeichnung, die das Restriktionensystem schwach respektiert, senkrecht ohne Knick gezeichnet werden können, wenn alle anderen Kanten ihre Knicke aus dem Algorithmus in Kapitel 6 behalten. □

Auch im Weiteren werden wir die Menge dieser Kanten mit E_s bezeichnen.

Genauso wie Kanten senkrecht ohne Knicke gezeichnet werden können, besteht auch die Möglichkeit, Kanten waagrecht ohne Knick zu zeichnen. Damit beschäftigen wir uns im Folgenden. Wir gehen wieder genauso vor wie bei den Kanten, die senkrecht ohne Knick gezeichnet werden können. Wieder zeigen wir die Eigenschaften nur für den Verlauf einer Kante, deren Lageabhängigkeiten festlegen, dass der linke Knoten immer oberhalb des rechten Endknotens liegt. Der andere Fall ist wieder analog bis auf eine Spiegelung.

Lemma 22

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E$ eine beliebige Kante aus der Kantenmenge. O.B.d.A. seien e \preceq_x und e \preceq_y so belegt, dass $d(e, v_1) = \searrow$ gilt.

Es gibt genau dann eine orthogonale planare Zeichnung Z des Graphen im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der e waagrecht ohne Knick gezeichnet wird, wenn

1. $e.pred(v_1) = s_r \wedge e.pred(v_2) = s_l$
2. Sei $e' = \{v_1, v_3\} = s_r.pred(v_1)$ der zweite Nachbar der Senkrechte auf der rechten Seite von v_1 . Falls e' in Z auf der rechten Seite von v_1 endet, dann muss gelten: Es existieren im Restriktionensystem keine Lageabhängigkeiten, die erzwingen, dass v_3 rechts von v_2 liegt.

BEWEIS Es ist die Äquivalenz der hier geforderten Eigenschaften mit den Eigenschaften aus Lemma 19 zu zeigen.

Eigenschaft 1 ist mit Eigenschaft 2 aus Lemma 19 und Eigenschaft ist 2 mit Eigenschaft 3 aus Lemma 19 äquivalent.

Also ist nur noch zu zeigen, dass die Eigenschaften aus diesem Lemma die Eigenschaften 1 und 4 aus Lemma 19 implizieren.

Eigenschaft 1 aus Lemma 19 ist erfüllt. Wenn es so einen Pfad gäbe, der v_1 und v_2 in y -Richtung ordnet, dann liegt er vollständig ober- oder unterhalb e und die erste Kante verlässt v_1 nach unten und die letzte verlässt v_2 nach oben. Dann kann aber e nicht, wie in Bedingung 1 gefordert, zu den beiden Senkrechten benachbart sein.

Auch Eigenschaft 4 aus Lemma 19 ist erfüllt. Es gibt nur dann keine planare Zeichnung, die das Restriktionensystem respektiert und in der v_3 links von v_2 liegt, wenn auf der Unterseite von v_3 eine Kante endet, deren zweiter Endknoten unterhalb von v_2 liegen muss. Angenommen es gibt eine Menge solcher Kanten. Sei nun v_4 der zweite Endknoten von derjenigen Kante aus dieser Menge, die zyklischer Vorgänger von e' am Knoten v_2 ist. Wenn es solche Kanten gibt, dann gibt es eine mit dieser Eigenschaft. Wegen unserer Kantenkonstruktion muss dieser zweite Endknoten v_4 dann aber links unterhalb v_3 liegen. Diese Kante liegt dann aber in der Fläche f , die die Kanten e und e' enthält. In diesem Fall liegt aber v_2 in Sektor 4 und v_3 in Sektor 3 und die Kanten zwischen den Knoten in diesen Sektoren ordnen v_3 und v_2 in x -Richtung, so dass $v_2 \preceq_x v_3$ gelten muss. Das wird aber nach Bedingung 2 ausgeschlossen. ■

Die Komplexität der Berechnung dieser Eigenschaften entspricht wieder der Komplexität der Berechnung der Kanten, die senkrecht ohne Knick gezeichnet werden können.

Lemma 23

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Dann kann für jede Kante in $O(1)$ bestimmt werden, ob sie die Verschiebeeigenschaften für eine waagerechte Kante ohne Knicke erfüllt.

BEWEIS Es müssen die Bedingungen aus Lemma 22 überprüft werden.

Die Bedingung eins kann direkt aus den Kantenattributen von e berechnet werden, indem wir die entsprechenden Nachbarkanten bei v_1 und v_2 überprüfen, ob dies die geforderten Senkrechten sind.

Für Bedingung zwei muss festgestellt werden, ob auf Grund des Restriktionensystems eine Lageabhängigkeit zwischen den Knoten v_2 und v_3 existiert, so dass v_3 rechts von v_2 liegen muss. Nach Lemma 6 existiert nur dann eine Lageabhängigkeit, wenn diese auch auf Grund von Kanten innerhalb einer gemeinsamen Fläche existiert. Dazu betrachten wir die Fläche, die e und e' gemeinsam haben. Eine Abhängigkeit existiert genau dann, wenn in f der Knoten v_3 den Anfang von Sektor 3 markiert. Dies kann in $O(1)$ aus den Attributen \preceq_x und \preceq_y der Kante $v_3.pred(e')$ berechnet werden. ■

Korollar 2

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Dann kann in $O(n)$ die Menge E_w aller Kanten bestimmt werden, die in einer orthogonalen planaren Zeichnung, die das Restriktionensystem schwach respektiert, waagrecht ohne Knick gezeichnet werden können, wenn alle anderen Kanten ihre Knicke aus dem Algorithmus in Kapitel 6 behalten. \square

Auch hier werden wir im Weiteren die Menge dieser Kanten mit E_w bezeichnen.

Die bisherigen Ergebnisse gelten unabhängig davon, ob eine Kante nach dem Basisalgorithmus mit einem oder zwei Knicken gezeichnet wird, da diese Eigenschaft in den Lemmata nicht berücksichtigt wird. Für Kanten, die bisher mit zwei Knicken gezeichnet werden, lassen sich bzgl. der Optimierung der Knickzahl weitere Eigenschaften herleiten. Eine dieser Eigenschaften zeigen wir hier, da sie für das folgende Kapitel von Bedeutung ist.

Für Kanten, die auf Grund der Struktur des Restriktionensystems bei Zeichnungen mit dem Basisalgorithmus zwei Knicke haben, kann man zeigen, dass solche Kanten nie senkrecht ohne Knick gezeichnet werden können.

Lemma 24 (Eine Kante aus E_2 kann nie senkrecht gezeichnet werden)

Gegeben sei ein Graph und ein legales Restriktionensystem. Dann kann keine Kante, die Element von E_2 ist, senkrecht ohne Knick gezeichnet werden.

BEWEIS Wir betrachten nur den Fall, dass die Kante mit zwei Knicken nach rechts unten verläuft. Der andere Fall ist symmetrisch.

Sei $e = \{u, v\} \in E_2$ eine Kante mit linkem Endknoten u und sei $d(e, u) = \searrow$. Wir zeigen, dass die Kante e am rechten Endknoten v nicht Nachbar der Senkrechte s_o sein kann, was eine der Voraussetzungen ist, damit die Kante senkrecht ohne Knick gezeichnet werden kann.

Wenn die Kante e Element von E_2 ist, dann gilt für ihren zyklischen Nachfolger e' am rechten Endknoten v , dass $d(e', v) = \swarrow$ (vgl. Regel eins und drei in Lemma 7). Damit ist e nicht Nachbar der Senkrechten s_o . \blacksquare

Wir haben damit alle Vorarbeiten geleistet, um im nächsten Schritt eine in der Summe der Kantenknicke optimale Lösung zu berechnen.

7.4 Statische Optimierung

Die Idee zur Optimierung beruht auf einer Transformation des Ausgangsproblems in einen Optimierungsgraphen $G_{opt} = (V_{opt}, E_{opt})$, in dem jedem gesparten Knick ein Knoten entspricht. Kanten in diesem Graphen werden genau dann eingefügt, wenn die beiden, den Knoten entsprechenden Einsparungen, nicht gleichzeitig durchgeführt werden können. Damit wird sichergestellt, dass die Einsparungen von Knicken, die ausgewählt werden, gleichzeitig durchführbar sind. Wir beschreiben dazu zunächst, wann zwischen

zwei Knoten im Optimierungsgraphen eine Kante eingefügt werden muss. Wir stellen dabei sicher, dass der Graph bipartit ist. Danach berechnen wir ein „maximum independent set“ ([GJ83]) des neu konstruierten Graphen.

Bei „maximum independent set“ wird eine maximale Teilmenge der Knotenmenge gesucht, wobei je zwei Knoten der Teilmenge nicht durch eine Kante verbunden sein dürfen. Dieses Problem ist im Allgemeinen NP-vollständig ([GJ83], GT20), allerdings existiert für bipartite Graphen eine polynomiale Lösung. In [KM92] wird gezeigt, wie die Menge der „maximum independent set“ eines bipartiten Graphen in $O(n^2 \log n)$ berechnet werden kann.

In einem ersten Schritt betrachten wir nur die Möglichkeit, Kanten waage- oder senkrecht ohne Knick zu zeichnen. Das Verfahren wird dann in einem zweiten Schritt so erweitert, dass eine Kante, die Element von E_2 ist, anstatt mit zwei Knicken auch mit einem Knick gezeichnet werden kann.

7.4.1 Waage- und senkrechte Kanten ohne Knicke

Wir berücksichtigen zunächst nicht, dass eine Kante mit zwei Knicken unter Umständen auch mit nur einem Knick gezeichnet werden kann. Wir betrachten also nur die Optionen, dass eine Kante senkrecht oder waagrecht ohne Knick gezeichnet werden kann. Dies vereinfacht den Algorithmus wesentlich. Wir werden in Abschnitt 7.4.2 sehen, dass bei der zusätzlichen Berücksichtigung der obigen Konstellation keine wesentlichen neuen Ideen nötig sind, allerdings gibt es einige recht umständlich zu behandelnde Spezialfälle.

Im Folgenden geben wir zuerst die Konstruktion des Graphen $G_{opt} = (V_{opt}, E_{opt})$ an. Danach zeigen wir, dass G_{opt} bipartit ist.

Die Menge V_{opt} ist so definiert, dass jedem gesparten Knick im Graphen G ein Knoten in V_{opt} entspricht. Damit führt die Auswahl einer maximalen Anzahl an Knoten aus V_{opt} zur Minimierung der Summe der Kantenknicke in der Zeichnung von G .

Es ist zu beachten, dass eine Kante, die nach dem Algorithmus in Kapitel 5 zwei Knicke hat, auch waagrecht ohne Knick gezeichnet werden kann.¹² Damit wird mit dieser waagerechten Kante nicht nur ein Knick, sondern es werden zwei Knicke eingespart. Würde man diese Einsparung mit nur einem Knoten in G_{opt} modellieren, dann wäre das falsch. Deshalb werden in so einem Fall zwei Knoten eingefügt. Wir stellen außerdem sicher, dass beide Knoten zu einer identischen Knotenmenge adjazent sind. Wenn also der Algorithmus zur Lösung von „maximum independent set“ einen der beiden auswählt, dann kann immer auch der andere gewählt werden, denn beide Knoten haben die gleiche Adjazenzmenge. Um diesen Abschnitt übersichtlicher zu gestalten, werden wir diese doppelten Knoten im Folgenden ignorieren.

Für die Beschreibung des Algorithmus zur Berechnung von G_{opt} wählen wir folgende Vorgehensweise:

Wir definieren zunächst die Menge von Knoten V_{opt} für einen neuen Graphen $G_{opt} = (V_{opt}, E_{opt})$. Danach definieren wir einige Kantenmengen im Graphen G_{opt} , wobei jede

¹² Aber nach Lemma 24 nicht senkrecht ohne Knick.

Kantenmenge eine Menge von Eigenschaften sicherstellt, die für eine planare Zeichnung erfüllt sein müssen. Für jede der definierten Mengen zeigen wir, mit welcher Komplexität und welchem Algorithmus diese Menge berechnet werden kann. Im Anschluss beschreiben wir dann den Algorithmus zur Berechnung von G_{opt} , der dann nur noch aus der sukzessiven Berechnung der eben definierten Mengen besteht.

Definition 15

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und die Mengen $E_s \subset E$ und $E_w \subset E$ aus den Korollar 1 und 2. Dann sei $G_{opt} = (V_{opt}, E_{opt})$ ein neuer Graph und die Menge der Knoten V_{opt} wie folgt definiert:

$$V_{opt} = \{v_{(e,w)} | e \in E_w\} \cup \{v_{(e,s)} | e \in E_s\}$$

Wir bezeichnen den ersten Index eines Knotens als *Kantenindex*. Er gibt an, für welche Kante der Knoten erzeugt wurde. Der zweite Index wird als *Richtungsindex* bezeichnet. Er gibt an, ob die Optimierung einer waage- oder einer senkrechten Kante ohne Knick entspricht. ■

Lemma 25

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und die Mengen $E_s \subset E$ und $E_w \subset E$ aus den Korollar 1 und 2. Die in Definition 15 definierte Kantenmenge V_{opt} des Graphen G_{opt} kann in $O(n)$ berechnet werden.

BEWEIS Konsequenz aus Korollar 1 und Korollar 2, indem für jede Kante aus den Mengen ein Knoten erzeugt wird. ■

Damit sind die Knoten des Optimierungsgraphen bestimmt. Es fehlen noch die Kanten, die gegenseitige Ausschlüsse von Einsparungen eines Knickes repräsentieren.

Es existieren Kanten, die sowohl waagrecht als auch senkrecht ohne Knick gezeichnet werden können. Es ist aber nicht möglich, beides gleichzeitig zu erreichen. Die folgende Definition charakterisiert eine Kantenmenge, die das sicherstellt.

Definition 16

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt. Dann sei $E_{WS} \subset E_{opt}$ die folgende Kantenmenge:

$$E_{WS} = \{\{v_{(e,w)}, v_{(e,s)}\} | e \in E \wedge v_{(e,w)}v_{(e,s)} \in V_{opt}\}$$

Lemma 26

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt. Die Menge E_{WS} kann in $O(n)$ berechnet werden.

BEWEIS Beim Erzeugen der Knotenmenge V_{opt} können zu jeder Kante $e \in E$ Verweise auf die Knoten $v_{(e,s)}$ und $v_{(e,w)}$ in G_{opt} angelegt werden, wenn diese Knoten existieren. Damit können die Kanten mit einem Durchlauf durch E in $O(n)$ berechnet werden. Wenn eine Kante e einen Verweis auf zwei Knoten $v_{(e,w)}$ und $v_{(e,s)}$ trägt, dann werden diese beiden Knoten durch eine Kante in G_{opt} verbunden. ■

Lemma 27

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt. Die Menge E_{WS} stellt sicher, dass eine Kante nie gleichzeitig waage- und senkrecht ohne Knick gezeichnet werden kann.

BEWEIS Die Knoten $v_{(e,w)}$ und $v_{(e,s)} \in V_{opt}$ repräsentieren die Kante e aus G . Da die Knoten in V_{opt} durch eine Kante verbunden sind, können sie nie gemeinsam in einem „maximum independent set“ ausgewählt werden. Somit kann e nur entweder senkrecht oder waagrecht gezeichnet werden. ■

Aus den Eigenschaften des Kandinsky-Modells ergeben sich zwei zusätzliche Gruppen von Kanten. Die erste stellt sicher, dass auf jeder Seite eines Knotens nur eine Kante ohne Knick enden kann.

Definition 17

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt. Dann sei $E_{BoE} \subset E_{opt}$ die folgende Kantenmenge:

$$E_{BoE} = \left\{ \left\{ v_{(e_1, x_1)}, v_{(e_2, x_2)} \right\} \left| \begin{array}{l} v_{(e_1, x_1)} \wedge v_{(e_2, x_2)} \in V_{opt} \wedge x_1 = x_2 \text{ und } e_1 \text{ und } e_2 \\ \text{enden an der gleichen Seite eines Knotens, wenn sie} \\ \text{ohne Knick mit Richtung } x_i \text{ gezeichnet werden.} \end{array} \right. \right\}$$

■

Lemma 28

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt. Die Menge E_{BoE} kann in $O(n)$ berechnet werden.

BEWEIS Wir durchlaufen die Kanten um jeden Knoten in G und überprüfen für je zwei aufeinander folgende Kanten, ob sie ohne Knick gezeichnet werden können und ob sie dann beide auf der gleichen Seite des Knotens enden. Falls dies gilt, wird die entsprechende Kante in G_{opt} eingefügt. Da auf diese Art jede Kante zweimal besucht wird, erfolgt die Berechnung in $O(n)$. ■

Lemma 29

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt. Die Menge E_{BoE} stellt sicher, dass an jeder Seite eines Knotens immer nur eine Kante ohne Knick enden kann.

BEWEIS Da die zwei Knoten in G_{opt} durch eine Kante verbunden werden, wenn die beiden Kanten ohne Kantenknick bei einer gemeinsamen Auswahl in einem „maximum independent set“ auf der gleichen Seite eines Knotens enden würden, wird eine gemeinsame Auswahl in einem „maximum independent set“ vermieden. ■

Zusätzlich muss im Kandinsky-Modell dafür gesorgt werden, dass jeder Kante ohne Knick ein Knick auf einer anderen Kante eindeutig zugeordnet werden kann. Wir wollen zunächst analysieren, unter welchen Umständen im 4-Sektorenmodell so ein Fall auftreten kann.

Um eine Konfliktsituation wie in Abbildung 7.11 zu erzeugen, muss in der Fläche, in der sich die Kanten schneiden, ein Sektor aus einer Kante bestehen und die beiden Kanten in den angrenzenden Sektoren können so ohne Knick gezeichnet werden, dass sie auf der gleichen Seite des Knotens beginnen wie die einzige Kante im Zwischensektor. Auch

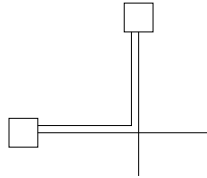


Abbildung 7.11: Anordnung von Kanten ohne Knicken führt zu einem Kantenschnitt.

diese Anordnungen müssen erkannt und durch eine Kante in G_{opt} verhindert werden.

Definition 18

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt. Dann sei $E_E \subset E_{opt}$ die folgende Kantenmenge:

$$E_E = \left\{ \left. \begin{array}{l} \{v_{(e_1,w)}, v_{(e_2,s)}\} \\ \exists f \in F : e_1 \text{ und } e_2 \text{ liegen in der Fläche } f \text{ und es liegt eine} \\ \text{weitere Kante } e \text{ in der Sequenz der Kanten um die Fläche} \\ \text{zwischen ihnen. Wenn } e \text{ mit einem Knick gezeichnet wird,} \\ \text{dann enden sowohl } e_1 \text{ und } e \text{ als auch } e_2 \text{ und } e \text{ auf der} \\ \text{gleichen Seite des gemeinsamen Endknotens.} \end{array} \right\} \right.$$

■

Lemma 30

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt. Die Menge E_E kann in $O(n)$ berechnet werden.

BEWEIS Bei einem Umlauf um jede Fläche im Uhrzeigersinn wird festgestellt, ob für je drei aufeinander Folgende Kanten die Bedingungen erfüllt sind. Dabei wird jede Kante zweimal besucht und die Komplexität liegt damit in $O(n)$. ■

Lemma 31

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt. Die Menge E_E stellt sicher, dass ein Knick auf einer Kante immer einer Kante ohne Knick eindeutig zugeordnet werden kann, wenn die beiden Kanten auf der gleichen Seite eines gemeinsamen Endknotens enden.

BEWEIS Wenn eine Kante zwei Knicke hat, dann ist die Bedingung immer erfüllt. Jeder der Knicke kann einer Kante ohne Knick an einem der Endknoten zugeordnet werden.

Wenn eine Kante e einen Knick hat, dann dürfen nicht gleichzeitig bei beiden Endknoten auf der gleichen Seite des Knotens Kanten ohne Knicke enden. Diese beiden Kanten ohne Knick müssen mit e in einer gemeinsamen Fläche f liegen. Damit es eine solche Konstellation überhaupt geben kann, muss die Kante e in einer solchen Fläche f die einzige in einem Sektor sein, denn andernfalls können die Vorgänger- und Nachfolgerkante nicht auf der gleichen Seite des gemeinsamen Endknotens enden. In einer derartigen Konstellation werden aber die beiden Knoten in V_{opt} , die den Kanten ohne Knick entsprechen, durch eine Kante verbunden und die Knoten können bei einer Berechnung von „maximum independent set“ nicht gemeinsam selektiert werden. ■

Damit haben wir Kantenmengen eingeführt, die sicherstellen, dass alle Anforderungen des Kandinsky Modells eingehalten werden.

Satz 9

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Sei die Menge der Knoten V_{opt} wie in Definition 15 festgelegt und sei $E_{opt} = E_{WS} \cup E_{BoE} \cup E_E$. Sei $V' \subseteq V_{opt}$ die Ergebnismenge der Berechnung von „maximum independent set“ für G_{opt} .

Wenn für den Graph G eine orthogonale Repräsentation nach dem Algorithmus in Abschnitt 6 berechnet wird und zusätzlich die Kanten gemäß der Auswahl V' ohne Knick gezeichnet werden, dann gibt es eine planare Zeichnung des Graphen mit dieser orthogonalen Repräsentation im Kandinsky-Modell.

BEWEIS In [Fös97] wird gezeigt, dass es für eine orthogonale Repräsentation eine planare Zeichnung eines Graphen im Kandinsky-Modell gibt, wenn die orthogonale Repräsentation sicherstellt, dass auf einer Seite eines Knotens immer nur eine Kante ohne Knick endet und jeder Kante e ohne Knick bei jeder anderen Kante, die mit e gemeinsam auf einer Seite eines Knotens endet, ein Kantenknick zugeordnet werden kann. Diese beiden Eigenschaften, zusätzlich mit der Einschränkung, dass eine Kante nie gleichzeitig senkrecht und waagrecht gezeichnet werden kann, werden durch die Kantenmengen sichergestellt. ■

Bis hier haben wir einen Graph konstruiert, bei dem jeder Knoten der Einsparung eines Knickes gegenüber dem Algorithmus zur Generierung einer orthogonalen Repräsentation aus Kapitel 6 entspricht. Außerdem haben wir in diesen Graph Kanten eingefügt, so dass je zwei Knoten, die zwei sich gegenseitig ausschließende Knickeinsparungen repräsentieren, durch eine Kante verbunden sind. Es bleibt zu zeigen, dass der generierte Graph bipartit ist, da andernfalls keine effiziente Lösung von „maximum independent set“ möglich ist. Wir zeigen das durch den Beweis, dass alle Zyklen im Graphen gerade Länge haben. Dies ist eine alternative Charakterisierung für bipartite Graphen (vgl. z. B. [Har74]).

Dabei gehen wir so vor, dass wir zunächst die Kantenmenge von G_{opt} so weit wie möglich einschränken und zeigen, dass der eingeschränkte Graph zyklensfrei ist. Danach erweitern wir die Kantenmenge schrittweise, bis schließlich G_{opt} vollständig ist.

Lemma 32

Gegeben sei ein Graph G mit einem legalen Restriktionensystem. Dann hat $G_{opt} = (V_{opt}, E_{BoE})$ keine Zyklen.

BEWEIS Eine Kante aus der Menge E_{BoE} verbindet immer nur Knoten in V_{opt} , die den gleichen Richtungsindex haben. Wir betrachten hier nur den Fall, dass der Richtungsindex w ist¹³, der andere Fall ist analog.

Der Beweis erfolgt über die Anordnung der Knoten in G durch die Kantenattribute \preceq_y (in Abbildung 7.12 ist die Beweisführung illustriert). Nehmen wir an, es gibt einen Zyklus in $G_{opt}|_{E_{BoE}}$. Seien dann die Knoten $v_{(e_1,w)}$ und $v_{(e_2,w)}$ Teil des Zyklus und seien diese Knoten durch eine Kante verbunden.

Wenn eine Kante zwei Knoten $v_{(e_1,w)}$ und $v_{(e_2,w)}$ verbindet, dann haben die beiden Kanten $e_1 = \{v_1, v\}$ und $e_2 = \{v_2, v\}$ in G einen gemeinsamen Endknoten, nämlich den Knoten v , an dem sie auf der gleichen Seite enden, wenn sie beide waagrecht ohne Knick gezeichnet werden. Beide Kanten müssen Nachbarn einer der beiden Senkrechten s_l oder s_r von v sein, einer der zyklische Vorgänger und einer der zyklische Nachfolger. Seien o.B.d.A e_1 und e_2 so, dass e_1 den Knoten v nach unten verlässt. Dann verlässt aber e_2 den Knoten v nach oben und es gilt, dass v_1 und v_2 in y -Richtung bzgl. \preceq geordnet sind, also $v_1 \preceq_y v_2 = true$.

O.B.d.A nehmen wir weiter an, dass es keinen Knoten v' in V gibt, so dass $v_2 \preceq_y v' = true$ gilt (v_2 ist also der oberste Endknoten der Kantenmenge, die den Zyklus in G_{opt} verursacht). Einen Knoten mit dieser Eigenschaft muss es geben, weil das System der Lagebeziehungen \preceq_y zyklisfrei ist.

Eine Voraussetzung für einen Zyklus ist eine zweifach Zusammenhangskomponente. Wenn die Knoten $v_{(e_1,w)}$ und $v_{(e_2,w)}$ Teil eines Zyklus sein sollen, dann muss sowohl der Knoten $v_{(e_1,w)}$ als auch der Knoten $v_{(e_2,w)}$ Endknoten einer weiteren Kante neben der Kante $\{v_{(e_1,w)}, v_{(e_2,w)}\}$ sein.

Wir betrachten den Knoten $v_{(e_2,w)}$. Es muss also neben e_1 eine weitere Kante geben, die nicht waagrecht gezeichnet werden kann, wenn e_2 waagrecht ohne Knick gezeichnet wird. Diese Kante verlässt dann aber v_2 nach oben, denn e_2 verlässt den Knoten v_2 nach unten. Daraus folgt aber, dass der zweite Endknoten v_3 von dieser Kante oberhalb von v_2 liegen muss, also $v_2 \preceq_y v_3 = true$. Das ist aber ein Widerspruch zu der Annahme, dass es keinen Knoten v' in V gibt, so dass $v_2 \preceq_y v' = true$ gilt.

Die Annahme ist also falsch und es kann keinen Zyklus geben. ■

Wir zeigen, dass zusätzliche Kanten in G_{opt} unter bestimmten Umständen keine ungeraden Zyklen erzeugen können. Das Lemma wird im Folgenden noch oft von Nutzen sein.

Lemma 33

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $G_{opt} = (V_{opt}, E_{opt})$ der daraus berechnete Optimierungsgraph. G_{opt} ist bipartit, falls für beliebige Paare $v_{(e_1,x)}$ und $v_{(e_2,y)}$ von Knoten aus G_{opt} , wobei $x, y \in \{w, s\}$ gilt, dass:

¹³ d.h. die Kante soll waagrecht ohne Knick gezeichnet werden

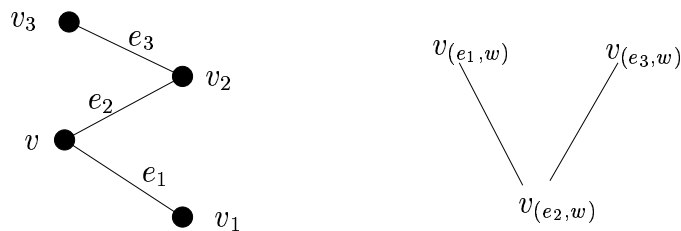


Abbildung 7.12: Ausschnitt aus einem Originalgraph und zugehöriger Ausschnitt aus dem Optimierungsgraph.

1. $x = y$, d. h. beide Optimierungen entsprechen einer waage- oder senkrechten Kante und
2. die Kanten e_1 und e_2 haben den gleichen Verlauf (von links oben nach rechts unten oder von links unten nach rechts oben) und
3. alle Pfade zwischen $v_{(e_1,x)}$ und $v_{(e_2,y)}$ haben gerade Länge;

oder

1. $x \neq y$, d. h. beide Optimierungen entsprechen nicht gleichzeitig einer waage- oder senkrechten Kante und
2. die Kanten e_1 und e_2 haben den gleichen Verlauf (von links oben nach rechts unten oder von links unten nach rechts oben) und
3. alle Pfade zwischen $v_{(e_1,x)}$ und $v_{(e_2,y)}$ haben ungerade Länge.

BEWEIS Ein Graph ist genau dann bipartit, wenn alle Zyklen gerade Länge haben. In einem Zyklus ist der „erste“ Knoten gleich dem „letzten“. Damit gilt, dass der Richtungsindex und der Verlauf der zugehörigen Kante gleich sind. Und in diesem Fall fordern wir, dass jeder Pfad gerade Länge hat, der Zyklus hat also gerade Länge.

Falls $x \neq y$ gilt, kann $v_{(e_1,x)}$ nicht gleich $v_{(e_2,y)}$ sein. Allerdings kann man einen Zyklus in zwei disjunkte Pfade $v_{(e_1,x)}, \dots, v_{(e_2,y)}$ und $v_{(e_2,y)}, \dots, v_{(e_1,x)}$ zerlegen. Da dann beide Pfade jeweils ungerade Länge haben, hat der Zyklus insgesamt gerade Länge. ■

Ein entsprechendes Lemma erhalten wir, wenn wir im zweiten Punkt jeweils fordern, dass der Kantenverlauf von e_1 ungleich dem Kantenverlauf von e_2 ist. Die Pfade müssen dann im ersten Fall ungerade und im zweiten Fall gerade Länge haben.

Im folgenden Lemma nehmen wir zu den Kanten aus E_{BoE} auch noch die Kanten aus E_{WS} hinzu.

Lemma 34

Gegeben sei ein Graph G mit einem legalen Restriktionensystem. Dann hat $G_{opt} = (V_{opt}, E_{BoE} \cup E_{WS})$ nur Zyklen mit gerader Länge.

BEWEIS Wir wissen aus Lemma 32 schon, dass bei alleiniger Betrachtung von Kanten aus E_{BoE} keine Zyklen, sondern nur Sequenzen in G_{opt} erzeugt werden. In so einer Sequenz gilt Lemma 33, da sich gegenseitig ausschließende waage- oder senkrechte Kanten immer einen unterschiedlichen Verlauf haben.

Kanten aus E_{WS} stellen Verbindungen zwischen diesen Sequenzen her. Die Knoten der einen Sequenz entsprechen waagerechten und die der anderen Sequenz entsprechen senkrechten Kanten. Bei der Verbindung der beiden Sequenzen trifft aber die zweite Gruppe von Bedingungen aus Lemma 33 zu. Denn die Kante in G_{opt} verbindet einen Knoten, der einer waagerechten, und einen Knoten der einer senkrechten Kante entspricht. Da es sich aber beide Male um die gleiche Kante handelt, ist der Kantenverlauf identisch und die Länge der Verbindung (eine Kante in G_{opt}) ist ungerade.

Damit ist der Graph bipartit, da für alle Sequenzen in G_{opt} die Bedingungen aus Lemma 33 zutreffen. ■

Schließlich nehmen wir noch die übrigen Kanten hinzu und erhalten den kompletten Graphen G_{opt} .

Lemma 35

Gegeben sei ein Graph G mit einem legalen Restriktionensystem. Dann ist $G_{opt} = (V_{opt}, E_{BoE} \cup E_{WS} \cup E_E)$ bipartit.

BEWEIS Nach Lemma 34 wissen wir, dass $G_{opt}|_{E_{BoE} \cup E_{WS}}$ nur gerade Zyklen hat. Es genügt also zu zeigen, dass der Graph G_{opt} auch unter Hinzunahme von E_E nur Zyklen gerader Länge hat.

Eine Kante aus E_E verbindet zwei Knoten in einer Situation wie in Abbildung 7.13. Wenn e_1 waagerecht und e_2 senkrecht gezeichnet wird, dann schneiden sich die beiden Kanten. So eine Situation kann lediglich entstehen, wenn wie in der Abbildung ein Sektor nur aus einer Kante besteht. Die beiden Kanten in den folgenden Sektoren können sich dann in der angegebenen Weise schneiden und die den Kanten entsprechenden Knoten werden in G_{opt} durch eine Kante aus E_E verbunden. Auf diese Verbindungskante treffen dann aber die Eigenschaften der zweiten Alternative aus Lemma 33 zu, da die beiden Kanten den gleichen Verlauf haben. ■

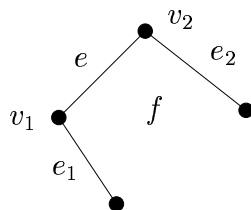


Abbildung 7.13: Anordnung, die eine Kante in E_E im Graphen G_{opt} erzwingt.

Nach diesen vorbereitenden Lemmata können wir das Haupttheorem dieses Abschnittes beweisen. Wir können einen Algorithmus angeben, der mit Hilfe eines Algorithmus für

„maximum independent set“ eine in unseren Voraussetzungen in der Knickzahl optimale Zeichnung des Graphen liefert.

Satz 10

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Dann liefert der folgende Algorithmus eine planare orthogonale Zeichnung des Graphen im Kandinsky-Modell mit minimaler Knickzahl, die das Restriktionensystem schwach respektiert, wenn die Möglichkeit ignoriert wird, dass eine Kante, die nach dem Algorithmus aus Kapitel 6 zwei Knicke hat, auch mit einem Knick gezeichnet werden kann:

Bestimme den Graphen $G_{opt} = (V_{opt}, E_{opt})$, indem zunächst die Knoten durch alle möglichen Knickeinsparungen berechnet werden (ohne Berücksichtigung der Möglichkeit, eine Kante mit einem statt mit zwei Knicken zu zeichnen). Eine Kante wird erzeugt, wenn

1. die beiden Knoten die gleiche Kante symbolisieren und der eine Knoten eine waage- und der andere eine senkrechte Kante symbolisiert,
2. die Bend-or-End Eigenschaft des Kandinsky-Modells verletzt wäre und zwei Kanten ohne Knick auf der gleichen Seite eines Knotens beginnen würden oder
3. sich zwei Kanten ohne Knick schneiden müssten, weil eine in der zyklischen Ordnung der Kanten um eine Fläche zwischen ihnen liegende Kante verhindern würde, dass eine der beiden Kanten vor ihrem Schnittpunkt endet.

Bestimme für $G_{opt} = (V_{opt}, E_{opt})$ ein „maximum independent set“ S .
Erzeuge eine orthogonale Repräsentation des Graphen, in der

1. eine Kante e eine Senkrechte ohne Knick ist, gdw. $\exists v_{(e,s)} \in S$,
2. eine Kante e eine Waagerechte ohne Knick ist, gdw. $\exists v_{(e,w)} \in S$,
3. der Verlauf alle übrigen Kanten wie im Basisalgorithmus erhalten bleibt.

BEWEIS Wir haben gezeigt, dass jede Auswahl eines „maximum independent set“ aus G_{opt} einer Menge von Einsparungen von Kantenknicken entspricht, die gemeinsam mit der Berechnung der übrigen Kantenverläufe nach dem Algorithmus aus Kapitel 6 eine orthogonale Repräsentation generiert, die den Anforderungen aus dem Satz entspricht. ■

Der letzte Satz dieses Abschnittes zeigt noch die Komplexität dieses Algorithmus.

Satz 11

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Dann gibt es einen Algorithmus, der eine planare orthogonale Zeichnung des Graphen im Kandinsky-Modell mit minimaler Knickzahl berechnet, die das Restriktionensystem schwach respektiert, wenn die Möglichkeit ignoriert wird, dass eine Kante, die nach dem Algorithmus aus Kapitel 6 zwei Knicke hat auch mit einem Knick gezeichnet werden kann. Der Algorithmus hat eine Laufzeit von $O(n^2 \log n)$.

BEWEIS Der Graph G_{opt} kann in Linearzeit berechnet werden (vgl. Lemma 25, 26, 28 und 30). Nach Lemma 35 ist G_{opt} bipartit. Um das „maximum independent set“ für G_{opt} zu berechnen, wird der Graph noch in seine Zusammenhangskomponenten zerlegt. Danach kann das „maximum independent set“ in $O(n^2 \log n)$ ([KM92]) berechnet werden. Die Berechnung aller übrigen Kantenverläufe erfolgt in $O(n)$, wie wir in Kapitel 6 gezeigt haben. Insgesamt erhalten wir also eine Laufzeit von $O(n^2 \log n)$. ■

Damit haben wir gezeigt, dass eine Knickoptimierung in polynomialer Zeit lösbar ist, wenn wir nur berücksichtigen, dass Kanten waage- oder senkrecht ohne Knick gezeichnet werden können. Wir müssen nun noch die Möglichkeit, eine Kante mit einem statt mit zwei Knicken zu zeichnen, in den bisherigen Ansatz integrieren.

7.4.2 Die Einsparung eines Knickes bei Kanten mit zwei Knicken

Bisher haben wir bei der Minimierung der Kantenknicke die Möglichkeit ignoriert, eine Kante mit einem anstatt mit zwei Knicken zu zeichnen.

Bei der Berechnung einer orthogonalen Repräsentation mit dem Algorithmus aus Kapitel 6 endet eine Kante auf der Ober- oder Unterseite des rechten Endknotens, außer wenn mit dieser Vorgabe keine planare Zeichnung möglich ist. In diesem Fall erhält die Kante einen zweiten Knick und endet auf der linken Seite des rechten Endknotens. Eine Kante, die auf diese Weise mit zwei Knicken gezeichnet wird, kann aber unter Umständen mit einem Knick gezeichnet werden, wenn sie auf der Ober- oder Unterseite des linken Endknotens beginnt. In Abbildung 7.5 ist dieser Zusammenhang dargestellt. Aufgabe dieses Abschnittes ist es also, die Kanten zu berechnen, bei denen das grundsätzlich möglich ist, und diese dann in G_{opt} zu integrieren.

Die Erweiterung des Algorithmus, der im vorhergehenden Kapitel entwickelt wurde, ist zwar aus Sicht der Ideen nicht schwer, allerdings bedingt durch viele Fallunterscheidungen komplex. Wir wollen die auftretenden Probleme an einem Einführungsbeispiel erklären.

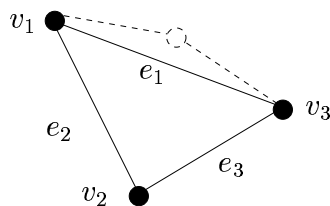


Abbildung 7.14: Einführungsbeispiel zur Einsparung eines Knickes bei Kanten mit zwei Knicken.

In Abbildung 7.14 ist ein kleiner Ausschnitt aus einem Restriktionensystem eines Graphen gegeben. Das Restriktionensystem sei so beschaffen, dass die Kante $e_1 \in E_2$ gilt. Die Kante e_1 kann nur dann mit einem Knick gezeichnet werden, wenn e_2 senkrecht ohne Knick gezeichnet wird, wenn die Zeichnung das Restriktionensystem schwach respektieren muss: Ansonsten würden sich e_1 und e_2 schneiden, weil e_1 mit einem Knick

den Knoten v_1 nach unten verlassen muss. Eine orthogonale Repräsentation, in der e_2 senkrecht und e_1 mit einem Knick gezeichnet ist, ist in Abbildung 7.15 dargestellt.

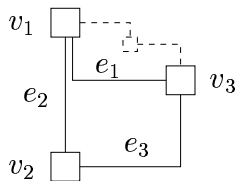


Abbildung 7.15: Einsparung eines Knickes bei e_1 hängt von einer Einsparung bei e_2 ab.

Diese Einsparungen sind erlaubt, denn bei beiden Kanten e_1 und e_2 wird ein Knick eingespart. Damit steht das Beispiel im Gegensatz zum bisherigen Ansatz, in dem stets gegolten hat, dass die Einsparung eines Knickes auf einer Kante von einer anderen Einsparung verhindert werden konnte. Hier ist eine Einsparung von einer anderen *abhängig*.

Weiter soll nun e_3 ebenfalls Element von E_2 sein. Nehmen wir einmal an, dass e_2 senkrecht ohne Knick gezeichnet wird. In diesem Fall erhalten wir folgende Einsparungsmöglichkeiten:

1. Sowohl e_1 als auch e_3 kann ohne Knick gezeichnet werden. In diesem Fall muss dann die jeweilig andere Kante beide Knicke behalten.
2. Die beiden Kanten e_1 und e_2 können gemeinsam mit je einem Knick gezeichnet werden. Dann würde zwar eine leere Fläche entstehen, was bei Kandinsky verboten ist. Wir wollen das aber explizit erlauben.

Insgesamt erhalten wir bei einer Modellierung wie im vorhergehenden Kapitel den in Abbildung 7.16 dargestellten Ausschnitt aus dem Optimierungsgraphen. Dabei steht die 1 anstelle des Richtungsindex für das Zeichnen einer Kante mit einem Knick. Wie man sieht, erhält man Zyklen ungerader Länge und der Optimierungsgraph ist damit nicht mehr bipartit. Damit kann das „maximum independent set“ nicht mehr in polynomialer Zeit berechnet werden.

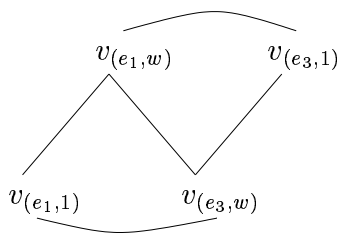


Abbildung 7.16: Optimierungsgraph zum Beispiel in Abbildung 7.14.

Und schließlich muss auch noch berücksichtigt werden, dass bei einer planaren Zeichnung der Knick einer Kante eindeutig einer Kante ohne Knick zugeordnet werden können



Abbildung 7.17: Wenn ein Knick auf einer Kante nicht eindeutig einer Kante ohne Knick zugeordnet werden kann, dann erhält man einen Kantenschnitt.

muss. In der Abbildung 7.17 ist auf der linken Seite ein Ausschnitt aus einem Restriktionensystem dargestellt. Nehmen wir an, dass e_1 ein Element von E_2 ist. Wenn e_1 mit einem Knick, e_3 senkrecht und e_2 waagrecht gezeichnet wird, dann schneiden sich e_2 und e_3 . Wenn nur zwei von diesen drei Optimierungen verwendet werden, dann ist eine planare Zeichnung möglich. Bisher war es immer so, dass sich zwei Optimierungen gegenseitig ausschließen. Eine direkte Modellierung einer solchen zwei aus drei Auswahl ist mit dem von uns gewählten Ansatz nicht möglich.

Das Ziel dieses Abschnittes ist es, die hier dargestellten zusätzlichen Probleme in das bisher beschriebene Verfahren zu integrieren. Zunächst wird untersucht, wann eine einzelne Kante mit einem anstatt mit zwei Knicken gezeichnet werden kann. Im zweiten Schritt wird dann der Graph G_{opt} so erweitert, dass er bipartit ist und die Auswahl eines „maximum independent set“ einer knickoptimalen Zeichnung entspricht.

Zeichnen von Kanten aus E_2 mit einem Knick

Das folgende Lemma beschäftigt sich mit der Frage, unter welchen Voraussetzungen eine Kante mit einem anstatt mit zwei Knicken gezeichnet werden kann.

Lemma 36

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E$ und gelte $e \in E_2$. O.B.d.A. sei v_1 der linke Endknoten von e und e verlaufe von links oben nach rechts unten¹⁴, also $d(e, v_1) = \searrow$. Weiter sei $e' = \{v_1, v_3\}$ der zyklische Nachfolger von e am Knoten v_1 .

Eine orthogonale planare Zeichnung im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der e mit nur einem Knick gezeichnet wird existiert genau dann, wenn weder

1. die Kante e' an der rechten Seite von v_1 endet noch
2. die Kante e' auf der Unterseite von v_1 endet, und gleichzeitig $d(e', v_1) = \searrow$ gilt und auf Grund der Lagebeziehungen des Restriktionensystems v_3 oberhalb von v_2 liegen muss.

¹⁴ Der Fall, dass e von links unten nach rechts oben verläuft, ist bis auf eine Spiegelung an der x-Achse identisch.

BEWEIS Wenn e mit einem Knick gezeichnet wird, dann endet e auf der Unterseite von v_1 .¹⁵

\Rightarrow : Die Kante e' ist der zyklische Nachfolger von e . Wenn e' auf der rechten Seite von v_1 endet, dann muss auch e auf der rechten Seite von v_1 enden, da e' andernfalls nicht der zyklische Nachfolger von e sein könnte.

Angenommen, e' würde ebenfalls auf der Unterseite von v_1 enden und es würde $d(e', v_1) = \searrow$ gelten. Wenn gleichzeitig v_3 oberhalb von v_2 liegen würde, dann würde die Kante e' die Kante e schneiden (siehe Abbildung 7.18), da die zyklische Ordnung der Kanten um die Knoten nicht geändert werden darf. Die Zeichnung könnte nicht planar sein.

\Leftarrow : Die Kante e muss auf der Unterseite von v_1 beginnen können, ohne dass die zyklische Ordnung der Kanten um die Knoten verletzt wird und ohne dass es zu einem Kantenschnitt kommt. Damit dies so ist, darf der zyklische Nachfolger von e nicht auf der rechten Seite von e enden, was durch Bedingung 1 sichergestellt ist.

Außerdem muss der Kantenknick von e näher am Knoten v_1 liegen als der von seinem zyklischen Nachfolger, da sich die Kanten sonst schneiden (siehe Abbildung 7.18). Das ist aber dadurch sichergestellt, dass nach Bedingung 2 in einer solchen Konstellation der Knoten v_3 nicht oberhalb von v_2 liegen muss. ■

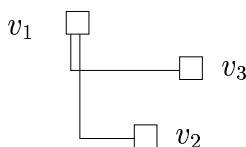


Abbildung 7.18: Illustration zum Beweis von Lemma 36.

Wir zeigen, wie und mit welcher Komplexität diese Bedingungen aus dem Restriktionensystem berechnet werden können.

Lemma 37

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E$ und gelte $e \in E_2$. O.B.d.A. sei $d(e, v_1) = \searrow$. Weiter sei $e' = \{v_1, v_3\}$ der zyklische Nachfolger von e am Knoten v_1 .

Eine orthogonale planare Zeichnung im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der e mit nur einem Knick gezeichnet wird, existiert genau dann, wenn in der Zeichnung weder

1. $d(e', v_1) = \searrow \wedge e' \in E_2$ und e' wird mit zwei Knicken gezeichnet.
2. $d(e', v_1) = \searrow \wedge e' \notin E_2$ und e' wird nicht senkrecht ohne Knick gezeichnet.

¹⁵ Ansonsten wäre die Planarität in der anderen Fläche, in der e liegt, verletzt. Das war ja genau der Grund, warum e zwei Knicke hatte.

3. $d(e', v_1) = \searrow$ und es existiert eine Fläche f , in der Sektor 1 und 3 leer sind, Sektor 2 nur aus der Kante e besteht und Sektor 4 mit Kante e' endet.

BEWEIS Es ist zu zeigen, dass die Bedingungen genau dann gelten, wenn die Bedingungen in Lemma 36 gelten.

\Leftarrow : Bedingungen 1 und 2 implizieren die Bedingung 2 von Lemma 36. Die Kante e' kann als zyklischer Nachfolger von e nur dann auf der rechten Seite von v_1 enden, wenn $d(e', v_1) = \searrow$ gilt. Wenn die Kante aber senkrecht ohne Knick gezeichnet wird bzw. mit einem anstatt mit zwei Knicken, dann endet sie auf der Unterseite von v_1 .

Die Bedingung 3 impliziert die Bedingung 2 von Lemma 36. Denn wegen Lemma 6 kann es eine solche Lagebeziehung nur dann geben, wenn es diese in einer gemeinsamen Fläche gibt. Die Lageabhängigkeit ergibt sich aber nur in der hier ausgeschlossenen Anordnung.

\Rightarrow : Die Kante e' darf nicht auf der rechten Seite von v_1 enden. Die Kante e' endet aber nur dann auf der rechten Seite von v_1 , wenn $d(e', v_1) = \searrow$ und e' wird wie mit dem Algorithmus aus Kapitel 6 gezeichnet. Deshalb muss, wie in Bedingungen 1 und 2 gefordert, eine Knickeinsparung bei e' durchgeführt werden, falls $d(e', v_1) = \searrow$ gilt.

Wenn es keine Lagebeziehungen geben darf, so dass v_3 oberhalb von v_2 liegen muss, dann darf es wegen Lemma 6 keine gemeinsame Fläche geben, die diese Lagebeziehung impliziert. Eine Anordnung wie in Bedingung 3 dieses Lemmas ist also nicht erlaubt. ■

Das folgende Lemma zeigt, dass die Überprüfung der Bedingungen für die Menge aller Kanten in $O(n)$ möglich ist.

Lemma 38

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Dann kann in $O(n)$ die Menge E_1 aller Kanten berechnet werden, die mit einem anstatt mit zwei Knicken gezeichnet werden können.

BEWEIS Es sind die Bedingungen aus Lemma 37 zu zeigen.

Die Bedingung 3 kann direkt aus den Attributen der Kanten e und e' und des zyklischen Vorgängers von e am Knoten v_2 berechnet werden, denn die Attribute \preceq_x und \preceq_y liefern die Sektoren der Kanten in f und zeigen damit, ob die Bedingung erfüllt ist.

Für Bedingung 1 und 2 muss überprüft werden, ob beim zyklischen Nachfolger von e an Knoten v_1 , der Kante e' , ebenfalls ein Knick eingespart werden kann. Bei dieser Kante kann es wieder von Bedingungen beim zyklischen Vorgänger abhängen usw. Die Frage kann aber trotzdem in Summe in Linearzeit beantwortet werden, wenn die Überprüfung der Bedingungen bei der Kante beginnt, die die letzte in der Sequenz der Kanten ist, die den Knoten v_1 nach rechts unten verlassen.

Wir müssen also bei der Überprüfung der Bedingungen mit der Kante e'' beginnen, für die $d(e'', v_1) = \searrow$ gilt und deren zyklische Nachfolger den Knoten v_1 nicht nach rechts unten verlässt. Damit kann man in einem Lauf um jeden Knoten die geforderten Bedingungen überprüfen. ■

Wir werden im Weiteren die Menge aller Kanten, die mit einem anstatt mit zwei Knicken gezeichnet werden können, mit E_1 bezeichnen.

Wir wollen noch einmal auf Lemma 37 eingehen. Nehmen wir an, dass der zyklische Nachfolger e' der betrachteten Kante e ebenfalls in E_2 liegt. In Punkt zwei wird gezeigt, dass e' ebenfalls mit einem Knick statt mit zwei gezeichnet werden muss, falls e' in die gleiche Richtung wie e geht. Die Einsparung eines Knickes auf Kante e ist also abhängig von der Optimierung bei e' . Um e' mit einem Knick zeichnen zu können, kann das gleiche Spiel mit dessen Vorgänger e'' nötig sein. Nur wenn bei e'' ein Knick eingespart wird, dann kann bei e und e' ein Knick eingespart werden.

Die Einsparung eines Knickes bei einer Kante kann also Einfluss auf die Einsparung eines Knickes bei einer Menge anderer Kanten haben. Mit diesen Mengen beschäftigen wir uns im folgenden Abschnitt genauer und zeigen, wie sie berechnet werden und wie sie in G_{opt} repräsentiert werden. Wir lösen damit das Problem, wie die Abhängigkeit einer Optimierung von einer anderen in unseren Lösungsansatz integriert werden kann.

Abhängigkeitsmengen

Wie wir gesehen haben, kann von der Einsparung eines Knickes bei einer Kante die Einsparung eines Knickes bei mehreren anderen Kanten abhängen. In der folgenden Definition führen wir für diese Kantenmenge den Begriff der Abhängigkeitsmenge ein.

Definition 19 (Abhängigkeitsmenge)

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e \in E$ eine Kante, die entweder senkrecht ohne Knick gezeichnet werden kann, oder als Element von E_2 mit einem Knick gezeichnet werden kann. Dann bezeichnet $E_{dep}(e)$ die Menge aller Kanten, bei denen die Einsparung eines Knickes nur dann möglich ist, wenn bei der Kante e eine der beiden obigen Knickeinsparungen vorgenommen wird.¹⁶ Wir bezeichnen $E_{dep}(e)$ als *Abhängigkeitsmenge* von e . ■

Nur Kanten, die Element von E_2 sind, können in einer Abhängigkeitsmenge liegen. Das folgt unmittelbar aus den Bedingungen, wann eine Kante senkrecht oder waagrecht ohne Knick gezeichnet werden kann. Dort gibt es keine Bedingung, nach der bei einer anderen Kante ein Knick eingespart werden muss.

Die Abbildung 7.19 zeigt einen Ausschnitt aus der orthogonalen Repräsentation¹⁷ eines Graphen. Dann erhalten wir für diesen Ausschnitt aus dem Graphen die folgenden Abhängigkeitsmengen:

- $E_{dep}(e_1) = \{e_2, e_3\}$
- $E_{dep}(e_2) = \{e_3\}$
- $E_{dep}(e_3) = \{\}$
- $E_{dep}(e_4) = \{e_5\}$

¹⁶ Das heißt, e wird senkrecht oder mit einem statt zwei Knicken gezeichnet.

¹⁷ Sie wird anstatt des Restriktionensystems dargestellt, da dann die Abhängigkeitsmengen leichter zu sehen sind.

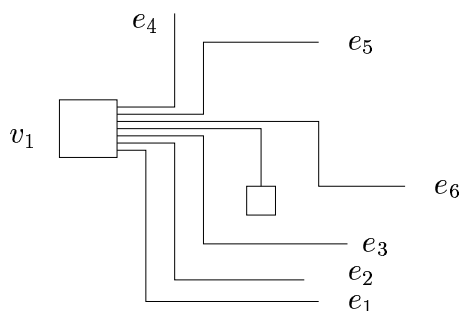


Abbildung 7.19: Abhängigkeitsmengen eines Graphen.

- $E_{dep}(e_5) = \{\}$

Man beachte, dass die Kante e_6 in keiner Abhängigkeitsmenge liegt. Ihr zyklischer Nachfolger an v_1 kann nicht senkrecht ohne Knick gezeichnet werden. Deshalb kann auch e_6 nicht mit einem Knick gezeichnet werden und $E_{dep}(e_6)$ ist leer.

Im Folgenden zeigen wir, wie E_{dep} berechnet werden kann. Danach kümmern wir uns um die Integration in G_{opt} . Wir beginnen mit dem Beweis, dass eine Abhängigkeitsmenge in konstanter Zeit in ihrer Größe berechnet werden kann.

Lemma 39

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e \in E$ eine Kante, die entweder senkrecht ohne Knick gezeichnet werden kann, oder als Element von E_2 mit einem Knick gezeichnet werden kann. Die Abhängigkeitsmenge der Kante e kann in konstanter Zeit in der Größe der Menge berechnet werden.

BEWEIS Gegeben sei eine Kante e , für die die Abhängigkeitsmenge berechnet werden muss. Sei v der linke Endknoten von e und o.B.d.A. gelte $d(e, v) = \searrow$.¹⁸

Wir laufen, beginnend bei Kante e , im Gegenuhrzeigersinn durch die Kanten um v , bis wir die erste Kante e' finden, die entweder nicht Element von E_2 ist oder v nicht nach rechts unten verlässt. Dann sind alle Kanten von e bis e' ausschließlich Element der Abhängigkeitsmenge von e . ■

Wir benötigen später noch die Eigenschaft, dass Abhängigkeitsmengen zyklensfrei sind. Das heißt, wenn eine Kante e Element der Abhängigkeitsmenge einer anderen Kante e' ist, dann kann e' nicht in der Abhängigkeitsmenge von e liegen.

Lemma 40

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $e \in E$ eine Kante aus der Kantenmenge und sei $e' \in E_{dep}(e)$ eine beliebige Kante aus der Abhängigkeitsmenge von e .

Dann gilt: $e \notin E_{dep}(e')$.

¹⁸ Der Fall $d(e, v) = \swarrow$ ist spiegelsymmetrisch bzgl. der x-Achse.

BEWEIS Sei e' Element der Abhängigkeitsmenge von e . Sei v der linke Endknoten von e und es gelte $d(e, v) = \searrow$. Wir müssen zeigen, dass e nicht Element der Abhängigkeitsmenge von e' sein kann.

Damit e' Element der Abhängigkeitsmenge von e sein kann, muss e' im Gegenuhrzeigersinn zwischen e und der Senkrechte s_r liegen. Wenn e Element der Abhängigkeitsmenge von e' wäre, dann müsste e im Gegenuhrzeigersinn zwischen e' und der Senkrechte s_r liegen. Beide Bedingungen können aber nicht gleichzeitig eintreten. Deshalb kann e nicht Element der Abhängigkeitsmenge von e' sein. ■

Eine Kante kann in der Abhängigkeitsmenge von mehreren anderen Kanten liegen. Deshalb ist die Größe der Menge aller Abhängigkeitsmengen nur durch $O(n^2)$ nach oben abschätzbar, nicht durch $O(n)$.

Allerdings müssen nicht alle Abhängigkeitsmengen berechnet werden. Wir werden zeigen, dass es genügt, die Abhängigkeitsmengen für maximal vier Kanten je Knoten zu berechnen. Die Ursache liegt in der Verwendung der Abhängigkeitsmengen. Sie werden benötigt, wenn ein Knoten in G_{opt} mit einem anderen durch eine Kante verbunden wird. Eine Kante in G_{opt} wird aber nur dann eingefügt, wenn sich zwei Optimierungen gegenseitig ausschließen. Also benötigen wir die Abhängigkeitsmenge nur für die Kanten, bei denen sich eine Einsparung eines Knickes mit einer anderen gegenseitig ausschließt.

Lemma 41

Gegeben sei ein Graph $G = (V, E)$ mit einem Restriktionensystem R . Dann kann es je Knoten v nur vier Kanten geben,

- die eine nicht leere Abhängigkeitsmenge haben können und
- bei denen die Einsparung eines Knickes sich gegenseitig mit der Einsparung eines Knickes bei einer beliebigen anderen Kante ausschließt, wenn die Einsparung des Knickes für die Einsparungen von Knicken bei den Kanten der Abhängigkeitsmenge nötig ist.

BEWEIS Eine Kante, die an einem Knoten v endet, kann nur dann eine nicht leere Abhängigkeitsmenge haben, wenn sie v nach rechts verlässt (vgl. Lemma 39). O.B.d.A. betrachten wir nur die Kanten, die v nach rechts unten verlassen.¹⁹

Es kann nur eine Kante geben, die v nach rechts unten verlässt und senkrecht ohne Knick gezeichnet werden kann. Diese Kante ist also eine Kante, die die obigen Bedingungen erfüllen kann. Ebenso kann es auch nur eine *erste* Kante im Gegenuhrzeigersinn geben, die v nach rechts unten verlässt, Element von E_2 ist, und mit einem Knick gezeichnet werden kann. Auch diese Kante kann die obigen Bedingungen erfüllen.

Bei allen anderen Kanten, die v nach rechts unten verlassen, kann die Einsparung eines Knickes, die zur Einsparungen bei den Kanten der Abhängigkeitsmenge nötig ist, keine gegenseitigen Ausschlüsse mit anderen Einsparungen eines Knickes haben. Wir zeigen diese Behauptung.

¹⁹ Die Überlegungen für die Kanten, die v nach rechts oben verlassen, ergeben sich durch eine Spiegelung der Überlegungen an der x-Achse.

Wenn die Einsparung eines Knickes für Einsparungen bei der Abhängigkeitsmenge nötig ist, dann muss bei dieser Einsparung die Kante immer an der Unterseite von v enden.

Entweder die Kante ist nicht Element von E_2 . Dann kann sie nicht senkrecht ohne Knick gezeichnet werden, denn es gibt nur eine Kante, die einen Knoten nach rechts unten verlässt und senkrecht ohne Knick gezeichnet werden kann, nämlich den Nachbarn der Senkrechten. Diese Kante haben wir aber oben schon zu den maximal vier Kanten gerechnet, die die Bedingungen erfüllen können.

Oder die Kante ist Element von E_2 . Beachte, dass ihr zyklischer Nachfolger ebenfalls Element von E_2 ist, denn andernfalls wäre sie die im Gegenuhrzeigersinn erste Kante, die v nach rechts unten verlässt. Auch diese Kante gehört aber zu den Kanten, für die wir die Abhängigkeitsmenge berechnen. Damit die Kante nicht mit einem Knick gezeichnet werden kann, muss ein Knoten oder eine Kante in dem in Abbildung 7.20 grau unterlegten Bereich liegen. Wegen des 4-Sektorenmodells und weil ihr zyklischer Nachfolger e' ebenfalls Element von E_2 ist, kann das nur e' sein und die Einsparung eines Knickes bei der Kante ist von einer anderen abhängig. Die Einsparung schließt sich aber mit keiner anderen gegenseitig aus.

Wir erhalten also nur maximal vier Kanten je Knoten, die die Bedingungen in dem Lemma erfüllen. Zwei beginnen wie eben gezeigt auf der Unterseite des Knotens, zwei durch eine Spiegelung der Überlegungen an der x-Achse an der Oberseite. ■

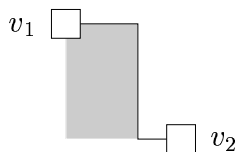


Abbildung 7.20: Illustration zum Beweis von Lemma 41.

Wir benötigen im Falle von einigen speziellen Modellierungen bei der Knickoptimierung das folgende Lemma, um die Korrektheit der Modellierung nachzuweisen. Es beschäftigt sich mit der Frage, welche Optimierungen bei einer Kante, deren Abhängigkeitsmenge nicht leer ist, möglich sind.

Lemma 42

Gegeben sei ein Graph $G = (V, E)$ mit einem Restriktionensystem R . Eine Kante e , deren Abhängigkeitsmenge nicht leer ist, kann nicht Element von E_w sein.

BEWEIS O. B. d. A. sei der Verlauf der Kante e von links oben nach rechts unten. Damit e waagrecht ohne Knick gezeichnet werden kann, muss ihr Vorgänger am linken Endknoten die Senkrechte s_r sein. Damit eine Kante in der Abhängigkeitsmenge von e liegt, muss sie am linken Endknoten zwischen e und der Senkrechten s_r liegen.

Es kann also e nicht gleichzeitig Element von E_w sein und eine nicht leere Abhängigkeitsmenge haben. ■

Die folgenden Überlegungen zeigen, wie welche Anforderungen eine Modellierung im Graph G_{opt} erfüllen muss, damit jede Auswahl eines „maximum independent set“ einer knickminimalen Zeichnung entspricht, wenn auch die Möglichkeiten zur Knickeinsparung in Abhängigkeitsmengen mit modelliert werden.

Wir definieren zunächst die Knotenmenge, die in den Optimierungsgraph alle Knoten einfügt, die das Zeichnen einer Kante mit nur einem anstatt mit zwei Knicken symbolisiert.

Definition 20

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Dann sei $V_1 \subset V_{opt}$ die folgende Knotenmenge:

$$V_1 = \left\{ v_{(e,1)} \left| \begin{array}{l} e \in E_2 \text{ und es existiert eine orthogonale Zeichnung,} \\ \text{die das Restriktionensystem schwach repektiert und} \\ e \text{ hat in der Zeichnung einen Knick.} \end{array} \right. \right\} \quad \blacksquare$$

Lemma 43

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und ein Graph $G_{opt} = (V_{opt}, E_{opt})$. Dann kann die Knotenmenge $V_1 \subset V_{opt}$ in $O(n)$ berechnet werden.

BEWEIS Konsequenz aus Lemma 38, denn dort wird gezeigt, dass die Menge der Kanten in G , die mit einem anstatt mit zwei Knicken gezeichnet werden können, in $O(n)$ berechnet werden kann. Damit kann ebenfalls in Linearzeit für jede dieser Kanten ein Knoten in G_{opt} erzeugt werden. ■

Diese neu zum Optimierungsgraph dazugenommene Knotenmenge muss noch mit Kanten zu den übrigen Knoten des Graphen verbunden werden, wenn sich zwei Einsparungen eines Knickes gegenseitig ausschließen. Wir geben hier nur an, welche Anforderungen an diese Kanten nötig sind, damit die Abhängigkeitsmengen korrekt berücksichtigt werden. Erst im nächsten Abschnitt geben wir die tatsächlich berechneten Kanten des Optimierungsgraphen an.

Lemma 44

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Sei $G_{opt} = (V_{opt}, E_{opt})$ der Graph, dessen Knoten- und Kantenmenge gemäß den Definitionen 15, 20, 16, 17 und 18 bestimmt ist.

Sei e eine Kante mit nicht leerer Abhängigkeitsmenge. Sei $v_{(e,x)}$ der Knoten in G_{opt} , der die Einsparung des Knickes symbolisiert, die Voraussetzung für die Einsparungen von Knicken bei den Kanten in $E_{dep}(e)$ ist. Sei V' die Menge aller Nachbarn von $v_{(e,x)}$ in G_{opt} .

Wenn für alle $e' \in E_{dep}(e)$ ihre zugehörigen Knoten in G_{opt} mit allen Knoten aus V' mit einer Kante verbunden werden, dann existiert für jede Auswahl eines „maximum independent set“ aus G_{opt} eine planare orthogonale Zeichnung des Graphen, die das Restriktionensystem schwach respektiert und in der alle Einsparungen von Knicken der Auswahl entsprechen.

Der Graph G_{opt} bleibt auch mit diesen Kanten bipartit, wenn er vorher bipartit war.

BEWEIS Jede Einsparung eines Knickes bei einer Kante der Abhängigkeitsmenge von e kann nur dann stattfinden, wenn bei e die dafür erforderliche Einsparung eines Knickes durchgeführt wird. Denn die Einsparung von Knicken bei e findet nicht statt, wenn ein zu $v_{(e,x)}$ adjazenter Knoten im „maximum independent set“ aus G_{opt} ausgewählt wird. Alle Knoten, die eine Optimierung einer Kante aus E_{dep} darstellen, sind aber mit den Nachbarn von v verbunden. Sie können also ebenfalls nicht ausgewählt werden.

Im Beweis zu Lemma 41 haben wir gezeigt, dass bei einer Kante aus einer Abhängigkeitsmenge sich das Zeichnen mit einem Knick nicht mit einer anderen Einsparung eines Knickes gegenseitig ausschließen kann. Damit verbinden diese zusätzlichen Kanten nur isolierte Knoten zum übrigen Graphen und der Graph bleibt bipartit. ■

Wir haben damit gezeigt, wie Abhängigkeitsmengen in G_{opt} repräsentiert werden müssen. Im Folgenden werden wir sie deshalb ignorieren und davon ausgehen, dass sie am Schluss repräsentiert werden, wenn alle sonstigen gegenseitigen Ausschlüsse berücksichtigt sind.

Bisher haben wir für Kanten aus E_2 , die mit einem Knick gezeichnet werden können, noch nicht angegeben, wie ihre zugehörigen Knoten in G_{opt} mit den übrigen Knoten durch Kanten verbunden werden müssen, wenn es gegenseitige Ausschlüsse mit anderen Optimierungen gibt. Die direkte Repräsentation ist leider nicht möglich. Wir haben das an dem einführenden Beispiel zu diesem Abschnitt der Arbeit bereits gezeigt. Im Folgenden zeigen wir, wie eine Integration trotzdem möglich ist, wobei wir allerdings eine Reihe von Fallunterscheidungen treffen müssen.

Ergänzung von G_{opt} , um Kanten aus E_2 mit einem Knick zu zeichnen

Die Ergänzung der Knotenmenge wurde bereits in der Definition 20 beschrieben. Im weiteren Verlauf müssen wir angeben, mit welchen anderen Knoten die neu erzeugten verbunden werden müssen, um gegenseitige Ausschlüsse mit anderen Optimierungsmöglichkeiten zu berücksichtigen. Wir haben die dabei entstehenden Probleme schon am Anfang des Kapitels beschrieben.

Wir lösen die Probleme, in dem wir für alle Situationen, in denen eine direkte Modellierung dazu führen würde, dass G_{opt} nicht mehr bipartit ist, jeweils in einem Spezialfall modellieren. Wir machen also Fallunterscheidungen und analysieren jeden Fall gesondert. Eine einfachere Vorgehensweise ist, wie die einführenden Beispiele gezeigt haben, nicht möglich. Um die Korrektheit der Modellierungen leichter zeigen zu können, benötigen wir noch die folgenden Lemmata.

Bei der Modellierung gehen wir oft so vor, dass eine bestimmte Optimierungsvariante auf jeden Fall verwendet wird. Das folgende Lemma zeigt, unter welchen Umständen eine knickoptimale Zeichnung existiert, in der eine bestimmte Kante senkrecht ohne Knick gezeichnet wird.

Lemma 45

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei $G_{opt} = (V_{opt}, E_{opt})$ der Graph, dessen Knoten- und Kantenmenge gemäß den Definitionen 15,

20, 16, 17 und 18 bestimmt ist. Sei $e = \{v_1, v_2\}$, $e \in E_s, e \notin E_w$ eine Kante mit linkem Endknoten v_1 .

Sei $e' = \{v_3, v_1\}$ eine Kante, die an Knoten v_1 endet, die Element von E_s ist und das senkrechte Zeichnen von e und e' schließen sich gegenseitig aus.

Alle anderen möglichen Knickeinsparungen werden dadurch, dass e senkrecht gezeichnet wird, nicht verhindert. Es gibt auch keine weitere Einsparung eines Knickes, die gemeinsam mit dem senkrechten Zeichnen von e eine dritte Einsparung eines Knickes verhindert.

Das heißt, $v_{e,s}$ hat Grad eins (oder null, wenn e' aus der ersten Bedingung nicht existiert).

Wenn alle Fälle zutreffen, dann gibt es eine knickminimale Zeichnung des Graphen G , die das Restriktionensystem schwach respektiert und in der die Kante e senkrecht ohne Knick gezeichnet wird.

BEWEIS Die Kante e kann nur senkrecht ohne Knick gezeichnet werden. Nehmen wir an, es gibt eine optimale Lösung, in der e nicht senkrecht ohne Knick gezeichnet werden kann. Wenn e nicht senkrecht ohne Knick gezeichnet wird, dann muss e' senkrecht ohne Knick gezeichnet sein. Denn das ist die einzige Einsparung eines Knickes, die Wechselwirkungen mit dem senkrechten Zeichnen von e hat.

Da e und e' nicht gleichzeitig senkrecht ohne Knick gezeichnet werden können, müssen sie beide zu einer gemeinsamen Senkrechten am Knoten v_1 benachbart sein.

Wir zeigen, dass die Abhängigkeitsmenge der Kante e' leer sein muss.

Da der Knoten v_1 der rechte Endknoten von e' ist, müssen Kanten aus der Abhängigkeitsmenge von e' am Knoten v_3 enden. O.B.d.A. sei $d(e, v_1) = \searrow$.²⁰ Damit gilt $d(e', v_1) = \swarrow$. Abbildung 7.21 stellt die Situation dar. Da sowohl e als auch e' Element von E_s sind, müssen die Knoten v_3 und v_2 in y-Richtung gegeneinander verschoben werden können. Deshalb muss es in der Fläche f unterhalb der beiden Kanten auch Elemente in den Sektoren 3 und 4 geben, sonst wären die beiden Knoten in y-Richtung geordnet.

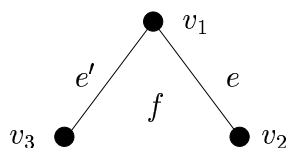


Abbildung 7.21: Bezeichnungen zu Lemma 45.

Also der Nachfolger von e' am Knoten v_3 die Richtung \swarrow oder \searrow . Kanten, die in der Abhängigkeitsmenge von e' liegen, müssen zwischen e' und s_r liegen und hätten damit die Richtung \swarrow . Da dies nicht der Fall sein kann, ist die Abhängigkeitsmenge von e' leer.

²⁰ Den anderen Fall erhält man durch Spiegelung an der x-Achse.

Da die Abhängigkeitsmenge von e' leer ist, ist es egal, ob e oder e' senkrecht ohne Knick gezeichnet wird und es gibt eine knickminimale Zeichnung des Graphen, die das Restriktionensystem schwach respektiert und in der e senkrecht ohne Knick gezeichnet wird. ■

Im folgenden Lemma vergleichen wir verschiedene Einsparmöglichkeiten für Kantenknicke in kleinen Ausschnitten des Graphen. Unter bestimmten Voraussetzungen ist eine Kombination von Einsparungen in diesem Ausschnitt immer besser als eine andere Kombination von Einsparungen für diesen Ausschnitt. Das heißt, man kann zeigen, dass nur eine der beiden Einsparungsvarianten berücksichtigt werden muss und trotzdem sichergestellt werden kann, dass eine knickminimale Zeichnung möglich ist. In der Modellierung in G_{opt} werden wir das verwenden, um manche Kombinationen von Einsparungen zu ignorieren.

Lemma 46

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Sei E' eine Teilmenge der Kantenmenge E und V' eine Teilmenge der Knotenmenge V , wobei V' die Menge der Endknoten von Kanten aus E' ist. Es gelte, dass für jede Kante $e \in E'$ auch alle Kanten aus der Abhängigkeitsmenge $E_{dep}(e)$ Element von E' sind. Außerdem seien zwei Optimierungsalternativen M_1 und M_2 bzgl. der Knickzahl für E' gegeben.

Für das Knickoptimierungsproblem für den Gesamtgraphen existiert eine minimale Lösung, in der die Alternative M_2 verwendet wird, falls gilt:

1. *Die Anzahl der Knicke für M_1 ist größer oder gleich der Anzahl der Knicke für M_2 und*
2. *Für alle $v \in V'$ gilt: Wenn auf einer Seite von v bei der Optimierungsvariante M_1 eine Kante aus $E \setminus E'$ ohne Knick enden kann, dann ist das auch bei Optimierung M_2 möglich. Das ist dann der Fall, wenn bei M_1 auf einer Seite des Knotens immer eine Kante ohne Knick endet, falls bei M_2 auf dieser Seite des Knotens eine Kante ohne Knick endet.*

BEWEIS Zunächst ist die Menge der Knicke im eingeschränkten Teilgraphen nicht größer, wenn M_2 statt M_1 in einer optimalen Lösung verwendet wird. Die zweite Bedingung stellt sicher, dass alle Optimierungsvarianten für den restlichen Graphen, die mit M_1 möglich sind, auch mit M_2 möglich sind. Damit kann es insgesamt nicht mehr Knicke geben, wenn M_2 statt M_1 verwendet wird.

Außerdem müssen auch die Abhängigkeitsmengen aller Kanten in E' liegen, so dass auch die möglichen Einsparungen eines Knickes, die nur bei Verwendung einer anderen Einsparung möglich sind, berücksichtigt sind. ■

In Abbildung 7.22 ist links ein Ausschnitt aus einem Graphen gegeben, der ohne die Einsparung von Knicken gezeichnet wurde. Hier kann auf Grund von Lemma 46 die Variante in der Mitte zu Gunsten der Variante rechts verworfen werden. Denn die Anzahl der Kantenknicke ist gleich und in der rechten Variante können auf allen Knotenseiten weitere Kanten ohne Knicke angeschlossen werden, wenn das in der Mitte möglich ist.

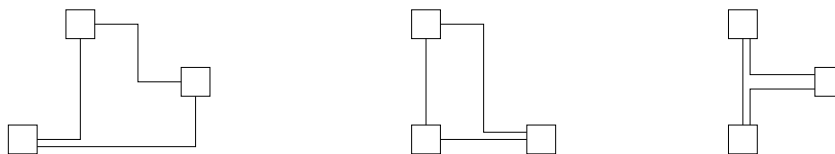


Abbildung 7.22: Statt der mittleren Optimierungsvariante kann immer die rechte verwendet werden.

Das Lemma 46 hilft also dabei, lokale Entscheidungen bei der Einsparung von Knicken zu treffen. Wir werden das oft nutzen, um auf diese Art bestimmte Kombinationen von Optimierungen zu verwerfen und statt dessen eine andere zu verwenden. Das benötigen wir, damit G_{opt} bipartit bleibt.

In den folgenden Abschnitten zeigen wir, wie die neuen Knoten mit Richtungsindex 1 im Graph G_{opt} integriert werden müssen, so dass jede ausgewählte Kombination von Optimierungen eine planare Zeichnung erlaubt. Außerdem muss es weiter jede Auswahl eines „maximum independent set“ einer knickoptimalen Zeichnung des Graphen G entsprechen. G_{opt} muss bipartit bleiben. Wir unterscheiden dabei verschiedene Fälle, die nacheinander bearbeitet werden.

Im Folgenden bezeichne stets $G_{opt} = (V_{opt}, E_{opt})$ den Optimierungsgraph, dessen Knoten- und Kantenmenge gemäß den Definitionen 15, 20, 16, 17 und 18 bestimmt ist.

Eine Kante aus E_2 kann nicht gleichzeitig mit einem und mit keinem Knick gezeichnet werden. Deshalb ergänzen wir die Kantenmenge von G_{opt} wie folgt, um das zu verhindern:

Definition 21

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und der Optimierungsgraph $G_{opt} = (V_{opt}, E_{opt})$. Dann sei

$$E_{1-w} = \{ \{v_{(e,w)}, v_{(e,1)}\} \mid \exists e : v_{(e,w)}, v_{(e,1)} \in V_{opt} \} \quad \blacksquare$$

Lemma 47

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und der Optimierungsgraph $G_{opt} = (V_{opt}, E_{opt})$. Die zusätzliche Kantenmenge E_{1-w} stellt sicher, dass keine Kante aus E_2 gleichzeitig mit einem und mit keinem Knick gezeichnet werden kann.

BEWEIS Kanten aus E_2 können nie senkrecht ohne Knick gezeichnet werden (vgl. Lemma 24). Damit muss nur verhindert werden, dass eine Kante aus E_2 gleichzeitig waagrecht und mit einem Knick gezeichnet werden soll. Dies verhindern die Kanten aus E_{1-w} . \blacksquare

Lemma 48

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und der Optimierungsgraph $G_{opt} = (V_{opt}, E_{opt})$. Wenn die Kantenmenge um E_{1-w} ergänzt wird, dann bleibt der Graph bipartit.

BEWEIS Einer der Endknoten der neuen Kanten, nämlich der Knoten mit Richtungsindex 1, war bisher in G_{opt} isoliert. Da jeder dieser Knoten maximal eine inzidente Kante erhält, bleibt der Graph bipartit. ■

Lemma 49

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und der Optimierungsgraph $G_{opt} = (V_{opt}, E_{opt})$. Die Menge E_{1-w} kann in $O(n)$ berechnet werden.

BEWEIS Wir speichern bei jeder Kante des Originalgraphen einen Verweis auf alle Knoten in G_{opt} , die für diese Kante erzeugt werden. Damit kann die Menge E_{1-w} mit einem Durchlauf durch die Menge der Kanten des Originalgraphen berechnet werden. ■

Im Rest dieses Kapitels beschäftigen wir uns mit der Frage, wann sich das Zeichnen einer Kante aus E_2 mit einem Knick mit der Einsparung eines Knickes bei einer anderen Kante gegenseitig ausschließt und wann nicht.

Zunächst sollen die Fälle herausgearbeitet werden, in denen es keine Wechselwirkungen mit anderen Einsparungen von Knicken gibt.

Fälle ohne Wechselwirkungen mit anderen Einsparungen von Knicken

Lemma 50

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E_1$ eine Kante mit linkem Endknoten v_1 . O.B.d.A. sei $d(e, v_1) = \searrow$ und sei e' der zyklische Nachfolger von e am Knoten v_1 .

Falls e' den Knoten v_1 nicht nach unten verlässt, dann kann e unabhängig von Knick-einsparungen bei allen anderen Kanten immer mit einem Knick gezeichnet werden.

BEWEIS Der Beweis ist in Abbildung 7.23 illustriert. Die Kante e kann wegen dem Verlauf von e' in keiner Abhängigkeitsmenge liegen und deshalb hängt das Zeichnen von e mit einem Knick von keiner anderen Einsparung eines Knickes ab.

Wir müssen noch untersuchen, ob es eine Einsparung eines Knickes geben kann, so dass ein Knoten in dem in Abbildung 7.20 schraffierten Bereich liegen muss, denn dann könnte e nicht mit einem Knick gezeichnet werden.

Mögliche Kandidaten dafür sind nur der zyklische Nachfolger an v_1 und der zyklische Vorgänger an v_2 . Die Kante e' verlässt den Knoten v_1 nach Voraussetzung nach oben. Betrachten wir also den Vorgänger $e'' = \{v_2, v_3\}$ von e am Knoten v_2 (siehe Abbildung 7.23 zur Illustration). Wenn er v_2 nach rechts verlässt, dann kann er nicht auf der linken Seite von v_2 enden, was aber nötig wäre, damit der Knoten in dem schraffierten Bereich liegen muss. Betrachten wir also den Fall, dass die Kante e'' den Knoten v_2 nach links verlässt. Da e' den Knoten v_1 nicht nach rechts unten verlässt, sind v_3 und v_1 in x -Richtung ungeordnet und v_3 kann links des schraffierten Bereiches in Abbildung 7.20 liegen. Wenn e'' den Knoten v_2 nach links oben verlässt, dann muss e'' ebenfalls Element von E_2 sein und der senkrechte Abschnitt von e'' kann ebenfalls links von diesem Bereich liegen. Damit gibt es keine Knoten oder Kanten, die in diesem Bereich liegen müssen.

Die Kante e kann also unabhängig von allen anderen Einsparungen mit einem Knick gezeichnet werden. ■

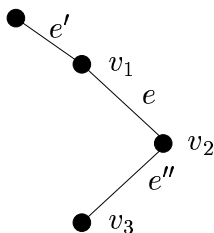


Abbildung 7.23: Bezeichnungen zum Beweis von Lemma 50.

Lemma 51

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Sei $e = \{v_1, v_2\} \in E_1$ eine Kante mit linkem Endknoten v_1 . O.B.d.A. sei $d(e, v_1) = \searrow$ und sei e' der zyklische Vorgänger von e am Knoten v_2 .

Falls e' den Knoten v_2 nicht nach links verlässt, dann verhindert das Zeichnen von e mit einem Knick keine Einsparung eines Knickes bei einer beliebigen anderen Kante.

BEWEIS Wir müssen untersuchen, ob ein anderer Knoten beim Zeichnen von e mit einem Knick in dem in Abbildung 7.20 schraffierten Bereich liegt.

Da die Kante e' den Knoten v_2 nach rechts verlässt, kann es bei v_2 keine Probleme geben. Sei $e'' = \{v_1, v_3\}$ der zyklische Nachfolger von e am Knoten v_1 (siehe Abbildung 7.24 zur Illustration). Wenn e mit einem Knick gezeichnet wird, dann kann das also nur

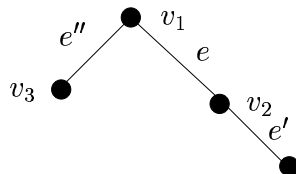


Abbildung 7.24: Bezeichnungen zum Beweis von Lemma 51.

dann noch die Einsparung eines anderen Knickes verhindern, wenn an der Unterseite von v_1 eine senkrechte Kante enden kann, deren zweiter Endknoten oberhalb von v_2 liegen muss. Weil aber die Kante e' den Knoten v_2 nach rechts verlässt, sind v_3 und v_2 in y -Richtung ungeordnet und v_2 kann oberhalb v_3 liegen. ■

Damit haben wir zwei Fälle gezeigt, bei denen keine weiteren Ergänzungen an G_{opt} nötig sind. Wir müssen im Folgenden also nur noch die Fälle untersuchen, die von den beiden Lemmata nicht abgedeckt werden.

Das Zeichnen einer Kante aus E_2 mit einem Knick und eine andere Optimierung schließen sich gegenseitig aus Im Folgenden sei stets $e_1 = \{v_1, v_2\} \in E_1$. Dabei sei v_1 der linke Endknoten von e_1 und $d(e_1, v_1) = \searrow$.²¹

²¹ Der Fall $d(e_1, v_1) = \nearrow$ ergibt sich durch eine Spiegelung an der x -Achse und wird deshalb hier nicht weiter beachtet.

Stets sei $e_2 = \{v_2, v_3\}$ der zyklische Vorgänger von e_1 am Knoten v_2 . Wegen Lemma 51 kann sich das Zeichnen von e_1 mit einem Knick nur dann mit einer anderen Einsparung eines Knickes gegenseitig ausschließen, wenn e_2 den Knoten v_2 nach links verlässt. Im Folgenden gelte also stets, dass e_2 den Knoten v_2 nach links verlässt.

Außerdem sei stets $e_3 = \{v_1, v_4\}$ der zyklische Nachfolger von e_1 am Knoten v_1 . Wegen Lemma 50 kann sich das Zeichnen von e_1 mit einem Knick nur dann mit einer anderen Einsparung eines Knickes gegenseitig ausschließen, wenn e_3 den Knoten v_1 nach unten verlässt. Im Folgenden gelte also stets, dass e_3 den Knoten v_1 nach unten verlässt.

Wir bezeichnen die Fläche, die von den drei Kanten begrenzt wird, mit f . Abbildung 7.25 illustriert sämtliche Bezeichnungen.

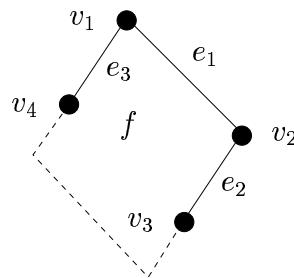


Abbildung 7.25: Bezeichnungen zu den folgenden Überlegungen.

Wir unterscheiden im Folgenden nach dem Verlauf der Kante e_2 . Der Verlauf von e_3 wird, wo nötig, im jeweiligen Unterabschnitt variiert. Die Tabelle in Abbildung 7.26 zeigt die unterschiedenen Fälle, gemeinsam mit dem Lemma, in dem die jeweilige Korrektheit der Modellierung für diese Situation gezeigt wird.

$d(e_2, v_2) = \swarrow$ Lemma 52			
$d(e_2, v_2) = \nearrow$	$e_2 \notin E_2$	$e_2 \notin E_w$ Lemma 53	
		$e_2 \in E_w$	v_1, v_3 in x-Richtung ungeordnet Lemma 54
			v_1, v_3 in x-Richtung geordnet Lemma 55
	$e_2 \in E_2$	v_1, v_3 in x-Richtung ungeordnet Lemma 56	
		v_1, v_3 in x-Richtung geordnet Lemma 57	

Abbildung 7.26: Verschiedene Fälle beim Zeichnen von Kanten mit einem anstatt mit zwei Knicken.

Beim Zeichnen von e_1 mit einem Knick kann es an zwei Stellen zu Wechselwirkungen

mit anderen Knickeinsparungen kommen. Wenn e_1 mit einem Knick gezeichnet wird, dann kann das Auswirkungen auf

1. das senkrechte Zeichnen von e_3 und
2. das waagerechte Zeichnen von e_2 oder das Zeichnen von e_2 mit einem Knick

haben. Zusätzlich muss noch beachtet werden, dass es nicht möglich ist, gleichzeitig e_1 mit einem Knick, e_3 senkrecht und e_2 waagerecht zu zeichnen.

Wir gehen bei der Untersuchung stets nach dem gleichen Schema vor: Zunächst wird die Situation genauer analysiert und dabei herausgearbeitet, welche zusätzlichen Bedingungen gelten müssen, damit an der bisherigen Modellierung in G_{opt} Änderungen vorgenommen werden müssen. Im zweiten Schritt wird dann gezeigt, wie diese Änderungen in G_{opt} aussehen. Es wird also gezeigt, mit welchen anderen Knoten ein Knoten mit Richtungsindex 1 verbunden werden muss, falls der betreffende Knoten in G_{opt} auch wirklich existiert. Danach zeigen wir, dass die Modellierung korrekt ist und dass G_{opt} bipartit bleibt.

Die folgende Definition verwenden wir, um die Anforderungen in den verschiedenen Fällen abzukürzen.

Definition 22 (legales Restriktionensystem)

Ein Restriktionensystem heißt *legales Restriktionensystem*, wenn der Graph G_{opt} bipartit ist und jede Auswahl eines „maximum independent set“ einer Zeichnung mit minimaler Knickzahl entspricht. ■

Die Kante e_2 verlässt Knoten v_2 nach links oben.

Lemma 52

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und eine Kante $e_1 = \{v_1, v_2\} \in E$. Sei $d(e_1, v_1) = \searrow$. Sei e_2 der zyklische Vorgänger von e_1 am Knoten v_2 und gelte $d(e_2, v_2) = \swarrow$. Sei e_3 der zyklische Nachfolger von e_1 am Knoten v_1 und e_3 verlasse den Knoten v_1 nach unten.

Dann gibt es eine Modellierung der Situation in einem legalen Restriktionensystem.

BEWEIS Nach Voraussetzung gilt $d(e_2, v_2) = \swarrow$. Falls $d(e_3, v_1) = \searrow$ gelten würde, dann wäre e_1 nicht Element von E_1 , da die Fläche f in diesem Fall nur aus den Sektoren 2 und 4 bestehen würde und als Folge müsste v_4 immer oberhalb von v_2 und rechts von v_1 liegen. Es muss also $d(e_3, v_1) = \swarrow$ gelten. Abbildung 7.27(a) illustriert die Situation. Es gilt:

1. Wegen e_2 ist e_1 am Knoten v_2 nicht Nachbar von der Senkrechte s_l und kann deshalb nicht waagerecht ohne Knick gezeichnet werden.
2. Der Knoten v_4 muss auf Grund der Lagebeziehungen oberhalb von v_2 liegen und deshalb kann e_3 nicht senkrecht gezeichnet werden, wenn e_1 mit einem Knick gezeichnet wird.

3. Der Knoten v_3 kann immer links von v_1 liegen. Deshalb ist jede Einsparung eines Knickes bei e_2 unabhängig vom Verlauf von e_1 .
4. Wegen e_1 muss auch e_2 Element von E_2 sein.

Abbildung 7.27(b) zeigt die Modellierung der Situation im Graphen G_{opt} . Hier ist auch berücksichtigt, dass e_2 mit einem Knick gezeichnet werden kann. (Unter der Annahme, dass auch e_2 Element von E_1 ist. Wenn dies nicht der Fall ist, dann würde der Knoten $v_{(e_2,1)}$ nicht existieren.)

Korrektheit der Modellierung: Alle bei der Optimierung nötigen Kombinationen können aus G_{opt} ausgewählt werden. Denn es wurden nur die beiden neuen Knoten $v_{(e_1,1)}$ und $v_{(e_2,1)}$ und je eine zu den Knoten inzidente Kante eingefügt. Diese Kanten ergeben sich aus den gegenseitigen Ausschlüssen mit anderen Optimierungen.

Alle wählbaren Kombinationen sind gültig, denn e_1 kann immer mit einem Knick gezeichnet werden, außer wenn e_3 senkrecht gezeichnet wird. Außerdem kann e_2 — bei ausschließlicher Betrachtung der hier untersuchten Fläche — immer mit einem Knick gezeichnet werden.²²

Wegen der Kante zwischen den Knoten $v_{(e_3,s)}$ und $v_{(e_1,1)}$ kann es nicht passieren, dass e_3 senkrecht, e_1 mit einem Knick und e_2 waagerecht gleichzeitig gewählt werden. Die Eigenschaft des Kandinsky-Modells, dass sich ein Knick bei einer Kante eindeutig einer Kante ohne Knick zuordnen lässt, ist also immer erfüllt.

G_{opt} **bleibt bipartit:** In den Graphen wird nur die zusätzliche Kante $\{v_{(e_3,s)}, v_{(e_1,1)}\}$ eingefügt. Der Knoten $v_{(e_1,1)}$ hat aber nur diese eine inzidente Kante, da die Kante e_1 nicht waagerecht ohne Knick gezeichnet werden kann. Der Graph bleibt bipartit. ■

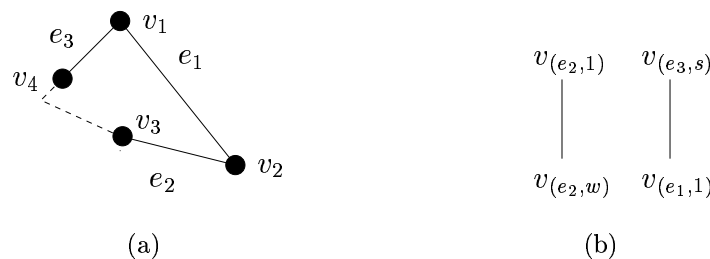


Abbildung 7.27: Modellierung, wenn $d(e_2, v_2) = \swarrow$ gilt.

Die Kante e_2 verlässt den Knoten v_2 nach links unten. Wir müssen hier verschiedene Fälle unterscheiden, da es keine gemeinsame Modellierung aller möglichen Fälle gibt.

Fall 1: Die Kante e_2 endet nicht auf der linken Seite von v_2 .

²² Sollten sich auf Grund der Anordnung der Fläche unterhalb von e_2 gegenseitige Ausschlüsse beim Zeichnen von e_2 mit einem Knick mit anderen Optimierungen ergeben, dann werden die entsprechenden Kanten beim Betrachten dieser Fläche in G_{opt} eingefügt.

Lemma 53

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und eine Kante $e_1 = \{v_1, v_2\} \in E$. Sei $d(e_1, v_1) = \searrow$. Sei e_2 der zyklische Vorgänger von e_1 am Knoten v_2 und gelte $d(e_2, v_2) = \swarrow$. Außerdem gebe es keine Zeichnung des Graphen im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der e_2 auf der linken Seite von v_2 endet. Sei e_3 der zyklische Nachfolger von e_1 am Knoten v_1 und e_3 verlasse den Knoten v_1 nach unten.

Dann gibt es eine Modellierung der Situation in einem legalen Restriktionensystem.

BEWEIS In diesem Fall kann e_2 weder Element von E_2 noch Element von E_w sein. Die Kante e_1 kann immer mit einem Knick gezeichnet werden. Denn v_4 kann in diesem Fall unterhalb von v_3 liegen, so dass gleichzeitig e_3 senkrecht und e_1 mit einem Knick gezeichnet werden können.

Wenn e_1 mit einem Knick gezeichnet wird, dann hat dies keinen Einfluss auf andere Knickeinsparungen und somit werden keine neuen Kanten in G_{opt} eingefügt. Der Graph G_{opt} bleibt damit bipartit. ■

Damit verbleiben die Fälle, in denen e_2 auf der linken Seite von v_2 enden kann. Die Kante e_2 muss also Element von E_w oder E_2 sein. Im Folgenden nehmen wir also immer an, dass e_2 entweder Element von E_2 oder von E_w oder von beidem ist.

Fall 2: Kante $e_2 \notin E_2 \wedge e_2 \in E_w$. v_3 und v_1 sind durch das Restriktionensystem in x-Richtung nicht geordnet oder v_1 muss rechts von v_3 liegen.

Lemma 54

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und eine Kante $e_1 = \{v_1, v_2\} \in E$. Sei $d(e_1, v_1) = \searrow$. Sei $e_2 = \{v_2, v_3\}$ der zyklische Vorgänger von e_1 am Knoten v_2 und gelte $d(e_2, v_2) = \swarrow$. Sei $e_2 \notin E_2 \wedge e_2 \in E_w$. Sei $e_3 = \{v_1, v_4\}$ der zyklische Nachfolger von e_1 am Knoten v_1 und e_3 verlasse den Knoten v_1 nach unten.

Gelte außerdem, dass v_3 und v_1 durch das Restriktionensystem in x-Richtung nicht geordnet sind oder dass v_1 rechts von v_3 liegen muss.

Dann gibt es eine Modellierung der Situation in einem legalen Restriktionensystem.

BEWEIS Es gilt:

1. Da v_1 und v_3 in x-Richtung gegeneinander verschoben werden können oder v_1 rechts von v_3 liegen muss, muss $d(e_3, v_1) = \swarrow$ gelten. Denn andernfalls müsste v_1 links von v_3 liegen. In Abbildung 7.28(a) sind die Knoten gemäß ihren Lagebeziehungen gezeichnet.
2. Bei einer ausschließlichen Betrachtung von e_1 und e_2 sieht man, dass e_1 mit einem Knick gezeichnet werden kann, unabhängig davon, ob e_2 waagerecht ohne Knick gezeichnet wird, da v_1 rechts von v_3 liegt oder liegen kann. An der bisherigen Modellierung müssen dafür keine Anpassungen vorgenommen werden.
3. Bei einer ausschließlichen Betrachtung von e_1 und e_3 sieht man, dass e_1 mit einem Knick gezeichnet werden kann, unabhängig davon, ob e_3 senkrecht ohne Knick gezeichnet wird. Denn es gilt:

- $d(e_3, v_1) \neq \searrow$
- In y-Richtung kann es keine Lageabhängigkeit von v_4 zu v_2 geben, denn es gilt $d(e_2, v_2) = \swarrow$. Deshalb kann v_4 unterhalb von v_2 liegen.

An der bisherigen Modellierung müssen dafür keine Anpassungen vorgenommen werden.

4. Wenn e_1 mit einem Knick, e_2 waagrecht und e_3 senkrecht gezeichnet wird, dann würde es einen Kantenschnitt geben.

Ein direkte Modellierung ist nicht möglich, da eine 2 aus 3 Auswahl nicht dargestellt werden kann. Deshalb müssen noch weitere Überlegungen angestellt werden.

Wenn e_3 senkrecht und e_2 waagrecht gezeichnet werden, dann muss e_1 mit zwei Knicken gezeichnet werden. Wegen Lemma 46 kann man das dadurch ersetzen, dass e_3 senkrecht und e_1 mit einem Knick gezeichnet werden, was in Abbildung 7.28(b) dargestellt ist.²³ Die Gesamtzahl an Knicken vergrößert sich dadurch nicht.

Wir wählen dann insgesamt die in Abbildung 7.28(c) dargestellte Modellierung der Situation in G_{opt} . Gegenüber der bisherigen Modellierung ist nur die Kante $\{v_{(e_3,s)}, v_{(e_2,w)}\}$ ergänzt worden.

Korrektheit der Modellierung: Als Erstes ist zu zeigen, dass alle bei der Optimierung erlaubten Kombinationen auch in G_{opt} ausgewählt werden können. Wir müssen also überprüfen, ob die durch Kante $\{v_{(e_3,s)}, v_{(e_2,w)}\}$ eingeführte Restriktion legal ist.

Mit dieser Kante ist es nicht möglich, gleichzeitig e_3 senkrecht und e_2 waagrecht zu zeichnen. Es wurde aber bereits angegeben, wie diese Kombination mit Hilfe von Lemma 46 ersetzt werden kann. Die zur Ersetzung gewählte Alternative, e_1 mit einem Knick und e_3 senkrecht zu zeichnen, kann aus G_{opt} ausgewählt werden, da die beiden entsprechenden Knoten nicht durch eine Kante verbunden sind. Damit können alle notwendigen Kombinationen von Knoten aus G_{opt} ausgewählt werden.

Zweitens ist zu zeigen, dass nicht erlaubte Kombinationen nicht ausgewählt werden können. Gegenüber Abschnitt 7.4.1 ist nur Knoten $v_{(e_1,1)}$ neu. Es müssen also nur Varianten überprüft werden, in denen $v_{(e_1,1)}$ vorkommt. Wie wir oben gesehen haben, muss als Einziges beachtet werden, dass e_3 senkrecht, e_2 waagrecht und e_1 mit einem Knick nicht gleichzeitig ausgewählt werden können. Das wird dadurch gewährleistet, dass die Kante $\{v_{(e_3,s)}, v_{(e_2,w)}\}$ neu hinzugekommen ist. Die Modellierung ist also korrekt.

G_{opt} **bleibt bipartit:** Neu ist nur die Kante $\{v_{(e_3,s)}, v_{(e_2,w)}\}$. Wegen Lemma 33 bleibt G_{opt} bipartit, da ein Knoten einer waagerechten und einer einer senkrechten entspricht und beide Kanten verschiedene Richtungen haben. ■

Fall 3: Kante $e_2 \notin E_2 \wedge e_2 \in E_w$. Wegen des Restriktionensystems gilt $v_1 \preceq_x v_3$.

²³ Der ersetzte Fall an sich bereitet keine Probleme. Allerdings kann seine Nichtbeachtung bei der Gesamtmodellierung ausgenutzt werden.

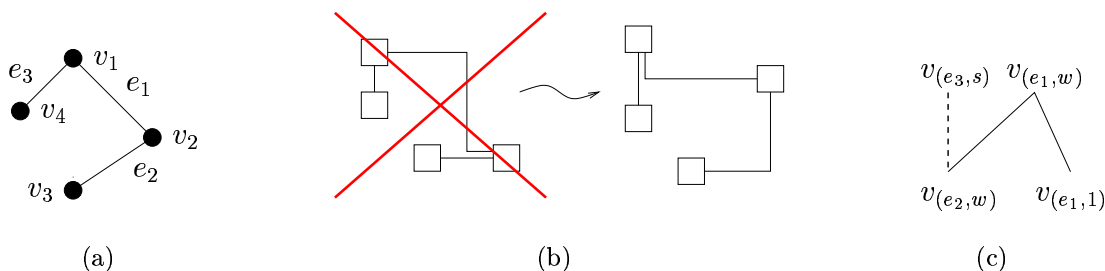


Abbildung 7.28: Illustration zu Fall 2 bei der Modellierung von Kanten aus E_2 in G_{opt} .

Lemma 55

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und eine Kante $e_1 = \{v_1, v_2\} \in E$ mit linkem Endknoten v_1 und sei $d(e, v_1) = \searrow$. Sei $e_2 = \{v_2, v_3\}$ der zyklische Vorgänger von e_1 am Knoten v_2 und gelte $d(e_2, v_2) = \swarrow$. Sei $e_2 \notin E_2 \wedge e_2 \in E_w$. Sei $e_3 = \{v_1, v_4\}$ der zyklische Nachfolger von e_1 am Knoten v_1 und e_3 verlasse den Knoten v_1 nach unten. Gelte außerdem, dass v_1 auf Grund des Restriktionensystems links von v_3 liegen muss.

Dann gibt es eine Modellierung der Situation in einem legalen Restriktionensystem.

BEWEIS Die Situation ist in Abbildung 7.29(a) dargestellt ist. Es gilt:

1. Da v_1 links von v_3 liegen muss, gilt $d(e_3, v_1) = \searrow$. In Abbildung 7.29(a) ist diese Lagebeziehung dargestellt.
2. Falls e_3 nicht senkrecht ohne Knick gezeichnet werden kann, dann kann e_1 nicht mit einem Knick gezeichnet werden. Dies ist ein Widerspruch zur Voraussetzung, dass $e_1 \in E_1$ gilt. Die Kante e_3 kann also senkrecht ohne Knick gezeichnet werden.
3. Nach Lemma 45 existiert eine knickoptimale Zeichnung, in der e_3 senkrecht gezeichnet wird. Denn am rechten Endknoten gibt es keine Optimierung, die dadurch verhindert werden kann. Denn es kann keine Kante geben, die alle Bedingungen erfüllt, um auf der Oberseite von v_4 ohne Knick zu enden. Entweder sie verlässt den Knoten nicht nach oben, oder ihr zweiter Endknoten muss wegen der Form von f unterhalb v_1 liegen.

Am linken Endknoten kann nur verhindert werden, dass eine andere Kante senkrecht ohne Knick gezeichnet wird.

4. Wenn die Kante e_3 senkrecht und e_2 waagerecht gezeichnet werden, dann muss e_1 mit zwei Knicken gezeichnet werden. Ansonsten hätte e_1 einen Knick und der Knick kann nicht einer Kante ohne Knick eindeutig zugeordnet werden.
5. Das Zeichnen von e_3 senkrecht und e_2 waagerecht kann nach Lemma 46 dadurch ersetzt werden, dass e_3 senkrecht und e_1 mit einem Knick gezeichnet werden (siehe Abbildung 7.29(b)).

Wir verwenden die Modellierung aus Abbildung 7.29(c). Gegenüber der bisherigen Modellierung müssen alle Knoten, die zu $v_{(e_3,s)}$ adjazent waren²⁴, aus G_{opt} entfernt werden. Knoten $v_{(e_2,w)}$ muss ebenfalls entfernt werden.

Korrektheit der Modellierung: Es kann eine knickoptimale Zeichnung ausgewählt werden, denn nach Lemma 45 existiert eine knickoptimale Lösung, in der e_3 senkrecht gezeichnet wird. Damit löschen wir alle Nachbarn von $v_{(e_3,s)}$ und e_3 wird senkrecht gezeichnet.

Knoten $v_{(e_2,w)}$ kann ebenfalls aus G_{opt} entfernt werden, da e_3 senkrecht immer ausgewählt wird und nach Lemma 46 wird die Kombination e_3 senkrecht und e_2 waagerecht durch eine andere Variante (e_3 senkrecht und e_1 mit einem Knick) ersetzt. Kante e_1 mit einem Knick schließt sich dann aber nur noch damit gegenseitig aus, dass e_1 waagerecht ohne Knick gezeichnet wird.

G_{opt} **bleibt bipartit:** Da nur Knoten und Kanten aus G_{opt} gelöscht werden, bleibt der Graph bipartit. ■

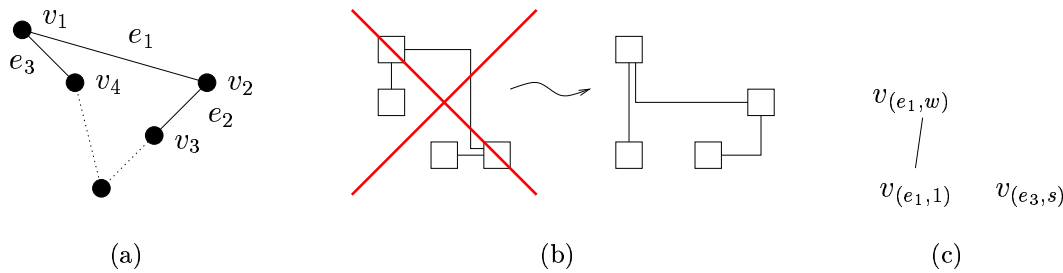


Abbildung 7.29: Illustration zu Fall 4 bei der Modellierung von Kanten aus E_2 in G_{opt} .

Fall 4: Die Kante e_2 ist Element von E_2 . v_3 und v_1 sind in x-Richtung ungeordnet.

Wir müssen hier zusätzlich eine weitere Kante mit in unsere Überlegungen einbeziehen. Für diesen Fall sei $e_4 = \{v_3, v_5\}$ der Vorgänger von e_2 an Knoten v_3 . In Abbildung 7.30(a) ist die Situation dargestellt.

Lemma 56

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und eine Kante $e_1 = \{v_1, v_2\} \in E$. Sei $d(e_1, v_1) = \searrow$. Sei $e_2 = \{v_2, v_3\}$ der zyklische Vorgänger von e_1 am Knoten v_2 und gelte $d(e_2, v_2) = \swarrow$. Sei $e_2 \in E_2$. Sei $e_3 = \{v_1, v_4\}$ der zyklische Nachfolger von e_1 am Knoten v_1 und e_3 verlasse den Knoten v_1 nach unten. Gelte außerdem, dass v_3 und v_1 durch das Restriktionensystem in x-Richtung nicht geordnet sind.

Dann gibt es eine Modellierung der Situation in einem legalen Restriktionensystem.

BEWEIS Sei $e_4 = \{v_3, v_5\}$ der zyklische Vorgänger von e_2 an Knoten v_3 . Es gilt:

²⁴ Da Lemma 45 anwendbar ist, wissen wir, dass das nur ein Knoten sein kann.

1. Die Forderung, dass die Knoten v_1 und v_3 gegeneinander verschoben werden können, impliziert, dass in der Fläche f sowohl Sektor 1 als auch Sektor 4 nicht leer sein dürfen. Sonst würde eine Lageabhängigkeit zwischen v_1 und v_3 vorliegen.
2. Wenn nur die Kanten e_1 und e_2 betrachtet werden, dann ergeben sich keine Probleme. Denn jede der Kanten kann unabhängig von der anderen Kante mit einem Knick gezeichnet werden. Also auch unabhängig davon, ob die jeweils andere Kante waagerecht gezeichnet wird.
3. Bei ausschließlicher Betrachtung von e_1 und e_3 sieht man, dass e_1 unabhängig von Optimierungen bei e_3 mit einem Knick gezeichnet werden kann, da v_4 unterhalb von v_2 liegen kann.
4. Bei ausschließlicher Betrachtung von e_2 und e_4 sieht man, dass e_2 unabhängig von Optimierungen bei e_4 mit einem Knick gezeichnet werden kann. Denn falls e_4 den Knoten v_3 nach links oben verlässt, dann kann v_5 oberhalb von v_2 liegen. Wenn e_4 den Knoten v_3 nach links unten verlässt, dann hat e_4 keinen Einfluss auf e_2 .
5. Es gibt keine planare Zeichnung, wenn e_1 mit einem Knick, e_2 waagerecht und e_3 senkrecht gezeichnet wird.
6. Falls e_4 den Knoten v_3 nach links oben verlässt, dann gibt es keine planare Zeichnung, falls e_1 waagerecht, e_2 mit einem Knick und e_4 senkrecht gezeichnet wird.
Falls e_4 den Knoten v_3 nicht nach links oben verlässt, dann muss die Kante den Knoten nach links unten verlassen. In diesem Fall kann dieses Problem also nicht auftreten.
7. Falls e_4 den Knoten v_3 nach links oben verlässt, und e_3 und e_4 senkrecht gezeichnet wird, dann muss entweder e_1 oder e_2 unabhängig vom Verlauf der anderen Kante (also e_1 oder e_2) mit zwei Knicken gezeichnet werden.
8. Nach Lemma 46 kann man e_2 waagerecht und e_3 senkrecht durch e_3 senkrecht und e_1 und e_2 mit einem Knick ersetzen (vgl. Abbildung 7.30(b) oben).
Falls e_4 den Knoten v_3 nach links oben verlässt, dann kann ebenso e_4 senkrecht und e_1 waagerecht durch e_4 senkrecht und e_1 und e_2 mit einem Knick ersetzt werden.
9. Falls e_4 den Knoten v_3 nach links oben verlässt, dann kann e_3 senkrecht und e_4 senkrecht dadurch ersetzt werden, dass nur eine der beiden Kanten senkrecht ohne Knick gezeichnet wird und dafür kann e_1 oder e_2 unabhängig vom Verlauf der anderen Kante mit einem statt mit zwei Knicken gezeichnet werden.

Wir wählen die Modellierung im Graphen G_{opt} , wie sie in Abbildung 7.30(c) dargestellt ist. Gegenüber der bisherigen Modellierung sind die Kanten $\{v_{(e_3,s)}, v_{(e_2,w)}\}$, $\{v_{(e_4,s)}, v_{(e_1,w)}\}$ und $\{v_{(e_3,s)}, v_{(e_4,s)}\}$ neu hinzugekommen.

Korrektheit der Modellierung: Wir zeigen zunächst, dass alle bei der Optimierung erlaubten Kombinationen auch ausgewählt werden können. Es sind also die Restriktionen durch die neuen Kanten zu untersuchen. Auch mit den Kanten $\{v_{(e_3,s)}, v_{(e_2,w)}\}$ und $\{v_{(e_4,s)}, v_{(e_1,w)}\}$ kann eine knickminimale Lösung ausgewählt werden. Wir haben oben angegeben, wie die entsprechenden Optimierungsalternativen ersetzt werden.

Die Kante $\{v_{(e_3,s)}, v_{(e_4,s)}\}$ verhindert, dass e_3 und e_4 gleichzeitig senkrecht gezeichnet werden. Wir haben oben angegeben, dass man das dadurch ersetzen kann, dass nur eine der beiden senkrecht und dafür aber entweder e_1 oder e_2 mit einem Knick anstatt mit zwei Knicken gezeichnet werden kann, was auch in G_{opt} gewählt werden kann.

Es ist noch zu zeigen, dass alle auswählbaren Varianten, die entweder $v_{(e_1,1)}$ oder $v_{(e_2,1)}$ enthalten, legal sind.

Die Kante $\{v_{(e_3,s)}, v_{(e_2,w)}\}$ verhindert, dass die Eigenschaft des Kandinsky Modells, dass ein Knick eindeutig einer Kante ohne Knick zugeordnet werden muss, verletzt wird, falls e_1 mit einem Knick gezeichnet wird. Genauso verhindert das die Kante $\{v_{(e_4,s)}, v_{(e_1,w)}\}$, falls e_2 mit einem Knick gezeichnet wird.

Zusätzlich ist es noch verboten, dass e_1 und e_2 mit einem Knick gezeichnet werden, wenn e_3 und e_4 senkrecht gezeichnet werden. Das wird aber durch die Kante $\{v_{(e_3,s)}, v_{(e_4,s)}\}$ verhindert. In allen anderen Kombinationen führt das Zeichnen von e_1 und e_2 mit einem Knick zu keiner Verletzung der Planarität, wie wir oben bereits erklärt haben.

G_{opt} **bleibt bipartit:** Alle drei neuen Kanten erfüllen die Voraussetzungen aus Lemma 33. Damit bleibt G_{opt} bipartit. ■

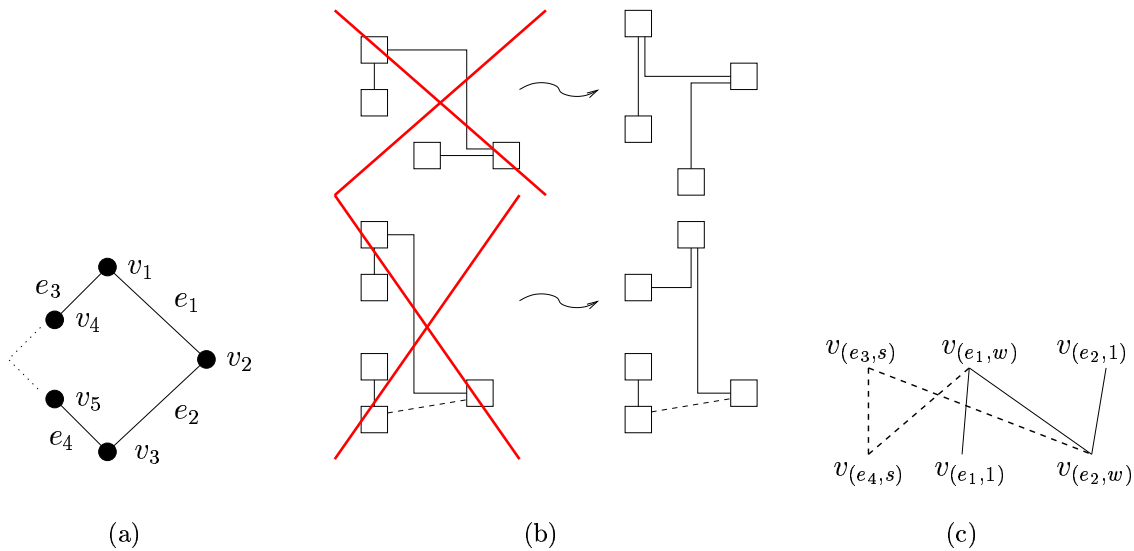


Abbildung 7.30: Illustration zu Fall 3 bei der Modellierung von Kanten aus E_2 in G_{opt} .

Fall 5: e_2 ist Element von E_2 . v_1 und v_3 sind in x-Richtung geordnet.

O.B.d.A. beschreiben wir nur den Fall, dass v_1 links von v_3 liegt.²⁵ Das hat aber zur Folge, dass wir auch Fälle berücksichtigen müssen, in denen $e_1 \notin E_1$ ist, e_2 aber schon. Wieder gibt es keine gemeinsame Modellierung aller möglichen Varianten und deshalb müssen wir hier drei Unterfälle unterscheiden.

Zunächst gilt, dass in der Fläche f Sektor 1 leer sein muss. Andernfalls müsste v_1 nicht links von v_3 liegen. Damit lassen sich die folgenden Fälle unterscheiden:

Die Fläche f besteht aus

1. e_1, e_2 und mehr als einer Kante in Sektor 4 (Abbildung 7.31(a)),
2. e_1, e_2 und mindestens zwei Kanten in Sektor 4 (Abbildung 7.34(a)),
3. e_1, e_2 und zusätzlichen Kanten in Sektor 3 und Sektor 4 (Abbildung 7.35(a)).

Wir werden diese Fälle im Rest des Abschnittes der Reihe nach abarbeiten.

Die Fläche f besteht aus e_1, e_2 und einer Kante in Sektor 4 (Abbildung 7.31(a)).

Lemma 57

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und eine Kante $e_1 = \{v_1, v_2\} \in E$. Sei $d(e_1, v_1) = \searrow$. Sei $e_2 = \{v_2, v_3\}$ der zyklische Vorgänger von e_1 am Knoten v_2 und gelte $d(e_2, v_2) = \swarrow$.

Außerdem gebe es eine Fläche f , die aus e_1 in Sektor 2, e_2 in Sektor 3 und einer Kante in Sektor 4 besteht.

Dann gibt es eine Modellierung der Situation in einem legalen Restriktionensystem.

BEWEIS Es gilt:

1. Falls e_3 nicht senkrecht gezeichnet werden kann, dann kann e_1 nicht mit einem Knick gezeichnet werden. Kante e_2 kann dann immer mit einem Knick gezeichnet werden, unabhängig von allen übrigen Optimierungen. Dieser Fall ist also unproblematisch, da nur der Knoten $v_{(e_2,1)}$ zu G_{opt} hinzugefügt und, falls vorhanden, mit dem Knoten $v_{(e_2,w)}$ verbunden werden muss. Der Graph G_{opt} bleibt bipartit.

Wir können also für den Rest des Falles nur die Situation untersuchen, dass e_3 senkrecht gezeichnet werden kann.

2. Auf Grund des Restriktionensystems gilt $e_1, e_2 \in E_1$.
3. Wenn neben e_3 senkrecht auch die Kante e_1 waagrecht gezeichnet wird, dann muss e_2 mit zwei Knicken gezeichnet werden. Nach Lemma 46 kann das dadurch ersetzt werden, dass e_3 senkrecht und e_1 und e_2 mit je einem Knick gezeichnet werden (vgl. Abbildung 7.31(b)).

²⁵ Der andere Fall ergibt sich durch eine Spiegelung an der x-Achse.

Wir wählen die in Abbildung 7.31(c) dargestellte Modellierung im Graphen G_{opt} , die zwei Besonderheiten aufweist. Ein Knoten mit Beschriftung $\neg v_{(e,x)}$ wird als Platzhalter für alle Nachbarn des Knotens $v_{(e,x)}$ verwendet. Und anstatt des Knotens $v_{(e_2,w)}$ existiert ein Knoten $v_{(e_2,1)'$. Wir interpretieren eine Auswahl aus G_{opt} wie folgt: Falls entweder $v_{(e_2,1)}$ oder $v_{(e_2,1)'}$ ausgewählt wird, dann wird die Kante e_2 mit einem Knick gezeichnet. Wenn beide Knoten ausgewählt werden, dann wird e_2 waagrecht ohne Knick gezeichnet.

Die Kante e_1 kann mit einem Knick gezeichnet werden, wenn entweder $v_{(e_1,1)}$ oder $v_{(e_1,1)'}$ aus G_{opt} ausgewählt wird. Unser Konzept ist es, dass alle Knoten, die Kanten aus der Abhängigkeitsmenge von e_1 repräsentieren, mit den Nachbarn von $v_{(e_1,1)}$ verbunden werden. Dieses Konzept wird durch $v_{(e_1,1)'}$ zerstört. Allerdings benötigen wir den Knoten $v_{(e_1,1)'}$ nur dann, wenn e_1 waagrecht gezeichnet werden kann. Wenn aber e_1 waagrecht gezeichnet werden kann, dann ist die Abhängigkeitsmenge der Kante leer (vgl. Lemma 42).

Korrektheit der Modellierung: Die Kante zwischen den Knoten $v_{(e_3,s)}$ und $v_{(e_1,w)}$ verhindert, dass die beiden zugehörigen Optimierungen gleichzeitig ausgeführt werden. Statt dessen können e_3 senkrecht und e_1 und e_2 mit einem Knick gezeichnet werden. Diese Einsparungen können in der Modellierung kombiniert werden.

Wenn e_2 waagrecht gezeichnet wird, dann muss e_1 immer mit zwei Knicken gezeichnet werden, wie es durch die gewählte Modellierung erzwungen wird. Denn falls e_3 nicht senkrecht gezeichnet wird, dann kann e_1 nicht mit einem Knick gezeichnet werden. Falls aber e_3 senkrecht gezeichnet wird, dann muss e_1 wieder mit zwei Knicken gezeichnet werden, da ein Knick bei e_1 der senkrechten Kante e_3 oder der waagerechten Kante e_2 zugeordnet werden muss.

Es können auch alle nötigen Einsparungen von Knicken kombiniert werden. Wir können gleichzeitig e_3 senkrecht und e_1 und e_2 mit einem Knick zeichnen, wir können e_1 waagrecht und e_2 mit einem Knick zeichnen.

G_{opt} **bleibt bipartit:** Auch mit der Kante zwischen den Knoten $v_{(e_3,s)}$ und $v_{(e_1,w)}$ bleibt G_{opt} nach Lemma 33 bipartit.

Durch die Kanten zwischen $v_{(e_1,w)}$ und $v_{(e_1,1)}$ und $v_{(e_1,1)}$ mit allen Nachbarn von $v_{(e_3,s)}$ werden Zyklen nur um Länge vier verlängert.

Knoten $v_{(e_2,1)'}$ wird wie Knoten $v_{(e_2,w)}$ verwendet. Dieser Knoten darf nach Lemma 33 mit Knoten $v_{(e_1,w)}$ durch eine Kante verbunden sein. Er darf damit auch mit allen Knoten, die eigentlich mit $v_{(e_2,w)}$ verbunden sein sollten, eine Kante gemeinsam haben. Damit ist G_{opt} bipartit. ■

Die Fläche f besteht aus e_1 , e_2 und mindestens zwei Kanten in Sektor 4 (Abbildung 7.34(a)).

Lemma 58

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und eine Kante $e_1 = \{v_1, v_2\} \in E$. Sei $d(e_1, v_1) = \searrow$. Sei $e_2 = \{v_2, v_3\}$ der zyklische Vorgänger von e_1 am Knoten v_2 und gelte $d(e_2, v_2) = \swarrow$. Außerdem gebe es eine Fläche f , die aus e_1 in Sektor 2, e_2 in Sektor 3 und mindestens zwei Kanten in Sektor 4 besteht.

Dann gibt es eine Modellierung der Situation in einem legalen Restriktionensystem.

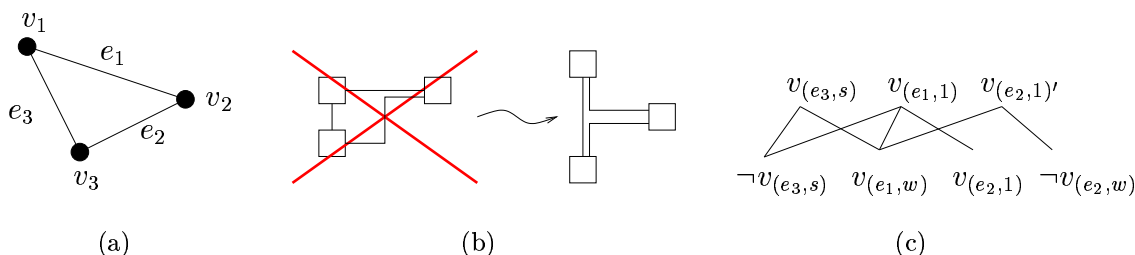


Abbildung 7.31: Illustration zu Fall 5, Variante 1 bei der Modellierung von Kanten aus E_2 in G_{opt} .

BEWEIS Sei e_3 der zyklische Nachfolger von e_1 am Knoten v_1 und e_4 der zyklische Vorgänger von e_2 am Knoten v_3 . Es gilt:

1. Angenommen, Kante e_3 kann nicht senkrecht gezeichnet werden. Dann ist e_1 auch nicht Element von E_1 und es muss noch überprüft werden, mit welchen Einsparungen von Knicken sich das Zeichnen von e_2 mit einem Knick gegenseitig ausschließt.
 - a) Wenn e_4 senkrecht und gleichzeitig e_1 waagerecht gezeichnet werden, dann kann e_2 nicht mit einem Knick gezeichnet werden.
 - b) Anstatt e_4 senkrecht und gleichzeitig e_1 waagerecht zu zeichnen, kann nach Lemma 46 e_1 waagerecht und e_2 mit einem Knick gezeichnet werden.

In diesem Fall verwenden wir die in Abbildung 7.34(c) dargestellte Modellierung.

2. Angenommen, Kante e_3 kann senkrecht gezeichnet werden. Dann ist e_1 auch Element von E_1 . Es gilt:
 - a) Nach Lemma 45 existiert eine knickoptimale Zeichnung, in der e_3 senkrecht ohne Knick gezeichnet wird. Denn am rechten Endknoten gibt es keine Einsparung eines Knickes, die dadurch verhindert werden kann. Am linken Endknoten kann nur verhindert werden, dass eine andere Kante senkrecht ohne Knick gezeichnet wird. In diesem Fall kann e_3 immer senkrecht gezeichnet werden.
 - b) In Abbildung 7.32 ist dargestellt, welche Optimierungen auf Grund von Lemma 46 durch andere ersetzt werden können.

Hier ist keine direkte Modellierung möglich und deshalb verwenden wir die in Abbildung 7.34(b) dargestellte. Wieder wird die spezielle Modellierung von Kante e_1 verwendet, bei der der Knoten $v_{(e_1,w)}$ durch den Knoten $v'_{(e_1,1)}$ ersetzt ist. Knoten $v'_{(e_1,1)}$ erbt alle Kanten von Knoten $v_{(e_1,w)}$.

Gegenüber der bisherigen Modellierung wurden außerdem alle Nachbarn von $v_{(e_3,s)}$ entfernt. Dafür erhalten wir die zusätzlichen Kanten $\{v_{(e_1,1)}, v_{(e_2,w)}\}$, $\{v_{(e_1,1)}, v_{(e_4,s)}\}$ und $\{v_{(e_1,1)'}, v_{(e_4,s)}\}$.

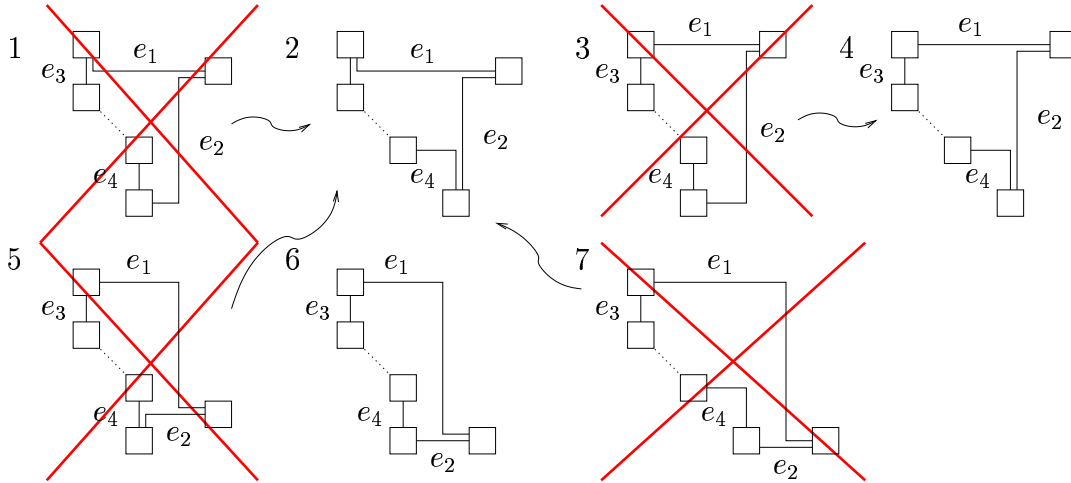


Abbildung 7.32: Ersetzung von Optimierungsmöglichkeiten nach Lemma 46.

Korrektheit der Modellierung: Wir betrachten zunächst den Fall, dass die Kante e_3 Element von E_s ist. Alle zur Optimierung nötigen Varianten können ausgewählt werden. Die Knotengruppen, die den Fällen zwei, vier und sechs in Abbildung 7.32 entsprechen, bilden je eine Menge von Knoten, die nicht durch eine Kante verbunden sind. Alle übrigen Alternativen können nach Lemma 46 durch eine andere ersetzt werden.

Jede Menge von Knoten, die $v_{(e_1,1)}$ oder $v_{(e_1,1)'}$ enthält und ausgewählt werden kann, entspricht einer planaren Zeichnung. Die Knoten $v_{(e_1,1)}$ und $v_{(e_1,1)'}$ können nur mit $v_{(e_2,1)}$ kombiniert werden, was immer dann legal ist, wenn e_4 nicht senkrecht gezeichnet wird. Die beiden Knoten haben aber eine Kante zu $v_{(e_4,s)}$. Jede kantenfreie Teilmenge, die $v_{(e_2,1)}$ enthält, entspricht ebenfalls einer planaren Zeichnung, denn e_2 kann mit einem Knick gezeichnet werden, wenn entweder e_4 nicht senkrecht oder e_1 mit zwei Knicken gezeichnet wird. Diese beiden Optimierungen schließen sich aber gegenseitig aus.

Für den Fall, dass e_3 nicht senkrecht gezeichnet werden kann, sind in Abbildung 7.33 alle Möglichkeiten dargestellt, wie Knicke eingespart werden können.

Nach Lemma 46 kann in Abbildung 7.33 Möglichkeit eins durch den zweiten Fall ersetzt werden. Die Situationen drei und vier sind unabhängig davon, wie Kante e_4 gezeichnet wird. Damit müssen die Situationen zwei, drei und vier modelliert werden. In Abbildung 7.34(c) ist eine entsprechende Modellierung angegeben. Gegenüber der bisherigen Modellierung ist die Kante $\{v_{(e_1,w)}, v_{(e_4,s)}\}$ neu hinzugekommen. Die Knoten, die den Bildern zwei, drei und vier in Abbildung 7.33 entsprechen, bilden jeweils eine unabhängige Menge in G_{opt} .

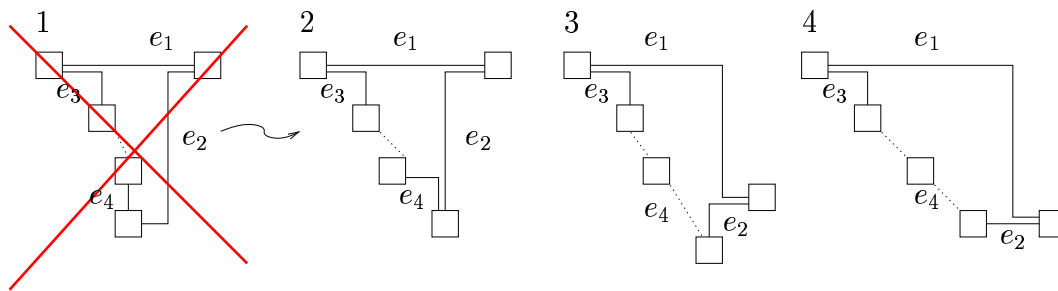


Abbildung 7.33: Einsparung von Knicken, wenn e_3 nicht Element von E_s ist.

Außerdem entspricht jede Kombination von unabhängigen Knoten, in der $v_{(e_2,1)}$ enthalten ist, einer planaren Zeichnung. Denn e_2 kann immer mit einem Knick gezeichnet werden, wenn nicht gleichzeitig e_4 senkrecht und e_1 waagrecht gezeichnet werden. Diese beiden Optimierungen schließen sich aber gegenseitig aus.

G_{opt} **bleibt bipartit:** Wir beschäftigen uns zunächst wieder mit dem Fall, dass e_3 senkrecht gezeichnet werden kann. Wir untersuchen die neuen Kanten. Es gilt, dass die Kante $\{v_{(e_1,1)'}, v_{(e_4,s)}\}$ alle Voraussetzungen für Lemma 33 erfüllt. Knoten $v_{(e_1,1)}$ hat genau zwei adjazente Kanten, die zwei Knoten durch einen Pfad mit gerader Länge verbinden. Die beiden Knoten waren bereits vorher durch einen Pfad mit gerader Länge verbunden und der Graph G_{opt} bleibt damit bipartit.

Schließlich muss noch die Modellierung für den Fall untersucht werden, wenn e_3 nicht senkrecht gezeichnet werden kann. Die neue Kante $\{v_{(e_1,w)}, v_{(e_4,s)}\}$ erfüllt alle Voraussetzungen für Lemma 33. Der Graph G_{opt} bleibt damit bipartit. ■

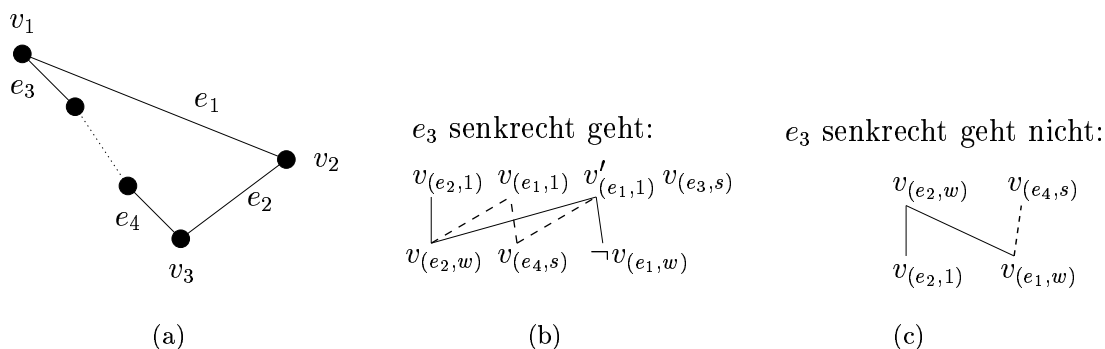


Abbildung 7.34: Illustration zu Fall 5, Variante 2 bei der Modellierung von Kanten aus E_2 in G_{opt} .

Die Fläche f enthält neben e_1 und e_2 weitere Kanten in Sektor 3 und Sektor 4 (Abbildung 7.35(a)).

Lemma 59

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und eine Kante $e_1 = \{v_1, v_2\} \in E$. Sei $d(e_1, v_1) = \searrow$. Sei $e_2 = \{v_2, v_3\}$ der zyklische Vorgänger von e_1 am Knoten v_2 und gelte $d(e_2, v_2) = \swarrow$. Außerdem gebe es eine Fläche f , die aus e_1 in Sektor 2, e_2 in Sektor 3 und zusätzlichen Kanten in Sektor 3 und 4 besteht.

Dann gibt es eine Modellierung der Situation in einem legalen Restriktionensystem.

BEWEIS Sei e_3 der zyklische Nachfolger von e_1 am Knoten v_1 und e_4 der zyklische Vorgänger von e_2 am Knoten v_3 . Es gilt:

1. Wenn die Kante e_3 nicht senkrecht ohne Knick gezeichnet werden kann, dann kann auch e_1 nicht mit einem Knick gezeichnet werden. Man erhält als zusätzliche Option ohne gegenseitigen Ausschluss mit einer anderen Einsparung eines Knickes, dass e_2 mit einem Knick gezeichnet werden kann. In diesem Fall genügt die bisherige Modellierung.
2. Wir gehen also im Weiteren davon aus, dass e_3 Element von E_s ist. Dann existiert nach Lemma 45 eine knickoptimale Zeichnung, in der e_3 senkrecht gezeichnet wird. Denn am rechten Endknoten gibt es wieder keine Einsparung eines Knickes, die dadurch verhindert werden kann. Wir können damit annehmen, dass e_3 senkrecht gezeichnet wird.
3. Wenn e_3 senkrecht und e_2 waagrecht gezeichnet wird, dann kann e_1 nicht mit einem Knick gezeichnet werden.
4. Die Optimierung e_3 senkrecht und e_2 waagrecht kann nach Lemma 46 dadurch ersetzt werden, dass e_3 senkrecht und e_1 und e_2 je mit einem Knick gezeichnet werden (vgl. Abbildung 7.35(b)).

Wir wählen für den Fall, dass e_3 senkrecht ohne Knick gezeichnet werden kann, die in Abbildung 7.35(c) dargestellte Modellierung in G_{opt} . Gegenüber der bisherigen Modellierung wurden alle Nachbarn von $v_{(e_3,s)}$ und der Knoten $v_{(e_2,w)}$ entfernt.

Korrektheit der Modellierung: Wir zeigen zuerst, dass alle notwendigen Teilmengen der Knotenmenge aus G_{opt} ausgewählt werden können. Das Entfernen aller Nachbarn von $v_{(e_3,s)}$ ist legal, da eine knickoptimale Zeichnung existiert, in der e_3 senkrecht gezeichnet wird. Auch $v_{(e_2,w)}$ kann entfernt werden, da e_3 senkrecht gezeichnet wird und angegeben wurde, wie die Kombination aus beidem durch eine andere Optimierung ersetzt werden kann, die auch aus G_{opt} ausgewählt werden kann.

Außerdem ist zu zeigen, dass jede Kombination, die gemeinsam mit $v_{(e_1,1)}$ oder $v_{(e_2,1)}$ ausgewählt werden kann, auch einer planaren Zeichnung entspricht. Das gilt aber immer, da das Auswählen von e_3 senkrecht und das Verhindern von e_2 waagrecht dafür sorgt, dass die beiden Kanten stets mit maximal einem Knick gezeichnet werden können.

G_{opt} **bleibt bipartit:** Es werden nur Objekte aus G_{opt} entfernt. Der Graph ist damit weiterhin bipartit. ■

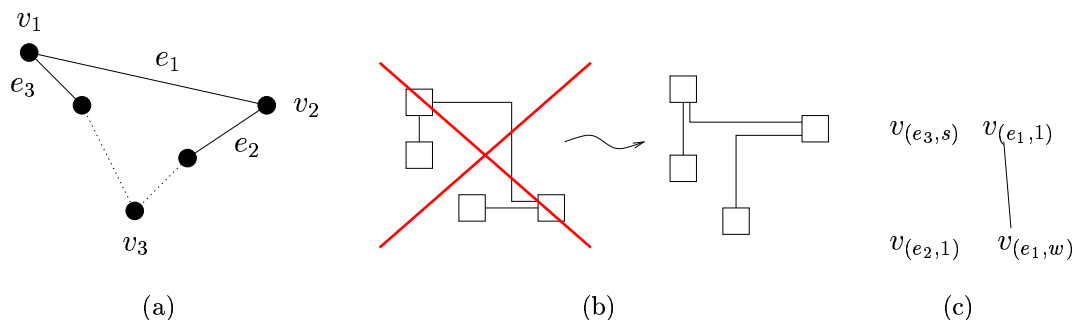


Abbildung 7.35: Modellierung in Fall 5, wenn Sektor 3 und 4 mehrere Knoten enthalten.

7.4.3 Zusammenfassung

Damit haben wir alle Fälle abgearbeitet. Das folgende Lemma fasst die Ergebnisse zusammen.

Lemma 60

Gegeben sei ein Graph G mit einem legalen Restriktionensystem R . Sei $e_1 = \{v_1, v_2\}$ Element von E_1 .

Es gibt eine Modellierung der Situation in G_{opt} , die diese zusätzliche Möglichkeit zur Einsparung eines Knickes berücksichtigt. Dabei bleibt der Graph G_{opt} bipartit und jede Auswahl eines „maximum independent set“ entspricht einer Zeichnung mit minimaler Knickzahl. Die Modellierung kann in $O(n)$ berechnet werden.

BEWEIS Folgerung aus den obigen Lemmata 47 und 50 bis 59.

Sei v_1 der linke Endknoten von e_1 . In 47 wird verhindert, dass eine Kante aus E_2 nicht gleichzeitig waagrecht und mit einem Knick gezeichnet werden soll. In Lemma 50 werden die Fälle behandelt, in der der zyklische Nachfolger e_3 bei v_1 den Knoten nicht nach unten verlässt, in Lemma 51 wird der Fall behandelt, wenn der zyklische Vorgänger bei v_2 den Knoten nicht nach links verlässt.

In Lemma 52 wird der Fall behandelt, dass e_2 den Knoten v_2 nach links oben verlässt und in den Lemmata 53 bis 59 die Fälle, in denen e_2 den Knoten v_2 nach links unten verlässt. Dabei wird jeweils noch mit berücksichtigt, ob e_3 den Knoten v_1 nach links unten oder rechts unten verlässt.

Dies sind alle möglichen geometrischen Anordnungen und jeweils wird eine Modellierung in G_{opt} gezeigt, die die geforderten Eigenschaften besitzt.

Zur Modellierung müssen immer nur Eigenschaften von wenigen ausgewählten Kanten betrachtet werden, so dass die Entscheidung für eine bestimmte Modellierung je Kante in konstanter Zeit möglich ist. Damit erhält man in Summe lineare Laufzeit. ■

Als Zusammenfassung aus diesem Abschnitt erhalten wir die folgenden Hauptaussagen dieses Kapitels. Sie zeigen, dass es einen Algorithmus gibt, der für unsere Anforderungen eine knickminimale Zeichnung erzeugt, wobei der Algorithmus eine Laufzeit von $O(n^2 \log n)$ besitzt.

Satz 12

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Dann liefert der folgende Algorithmus eine planare orthogonale Zeichnung des Graphen im Kandinsky-Modell mit minimaler Knickzahl, die das Restriktionensystem schwach respektiert:

Bestimme den Graphen $G_{opt} = (V_{opt}, E_{opt})$, indem zunächst die Knoten durch alle möglichen Knickeinsparungen berechnet werden. Eine Kante wird erzeugt, wenn

1. die beiden Knoten die gleiche Kante symbolisieren und der eine Knoten eine waage- und der andere eine senkrechte Kante symbolisiert,
2. die Bend-or-End Eigenschaft des Kandinsky-Modells verletzt wäre und zwei Kanten ohne Knick auf der gleichen Seite eines Knotens beginnen würden oder
3. sich zwei Kanten ohne Knick schneiden müssten, weil eine in der zyklischen Ordnung der Kanten um eine Fläche zwischen ihnen liegende Kante verhindern würde, dass eine der beiden Kanten vor ihrem Schnittpunkt endet.

Ergänze die Knoten- und Kantenmenge von G_{opt} um auch die Situationen zu berücksichtigen, wenn sich das Zeichnen einer Kante aus E_2 mit einem Knick und das Einsparen eines anderen Knickes gegenseitig ausschließen.

Bestimme für G_{opt} ein „maximum independent set“ S .

Erzeuge eine orthogonale Repräsentation des Graphen, in der

1. eine Kante e eine Senkrechte ohne Knick ist, gdw. $v_{(e,s)} \in S$,
2. eine Kante e eine Waagerechte ohne Knick ist, gdw. $v_{(e,w)} \in S$ oder $v_{(e,1)} \wedge v_{(e,1)'} \in S$,
3. eine Kante aus E_2 mit einem Knick gezeichnet wird, gdw. $v_{(e,1)} \vee v_{(e,1)'} \in S$. Diese Kanten beginnen dann auf der Ober- oder Unterseite des linken Endknotens und enden auf der linken Seite des rechten Endknotens und
4. der Verlauf alle übrigen Kanten wie im Basisalgorithmus erhalten bleibt.

BEWEIS Folgerung aus Satz 10 und Lemma 60. ■

Satz 13

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Dann gibt es einen Algorithmus mit Laufzeit $O(n^2 \log n)$, der eine planare orthogonale Zeichnung des Graphen im Kandinsky-Modell mit minimaler Knickzahl erzeugt, die das Restriktionensystem schwach respektiert.

BEWEIS Der Graph G_{opt} kann in Linearzeit berechnet werden (vgl. Lemma 25, 26, 28, 30 und 60).

Nach den Lemmata 35 und 60 ist G_{opt} bipartit. Um das „maximum independent set“ für G_{opt} zu berechnen, wird der Graph noch in seine Zusammenhangskomponenten zerlegt. Danach kann der Algorithmus zur Berechnung des „maximum independent set“ verwendet werden. Die Laufzeit dieses Algorithmus liegt in $O(n^2 \log n)$ (vgl. [KM92]). Die Berechnung aller übrigen Kantenverläufe erfolgt in $O(n)$, wie wir in Kapitel 6 gezeigt haben.

Insgesamt erhalten wir also eine Laufzeit von $O(n^2 \log n)$. ■

Wir haben damit gezeigt, dass man im 4-Sektorenmodell in polynomialer Zeit innerhalb der aus interaktiven Gesichtspunkten gewählten Schranken die Knickzahl optimieren kann. Die Konstruktion des Graphen G_{opt} in Linearzeit bietet einen Ansatzpunkt für schnelle Heuristiken. Es ist z.B. denkbar, für jede Zusammenhangskomponente in G_{opt} in Linearzeit die Knotenmenge in zwei Teilmengen zu zerlegen, wobei innerhalb jeder Knotenmenge keine zwei Knoten durch eine Kante verbunden sind. Die größere der beiden Mengen wird zum Ergebnis hinzugezählt.

Solange bei Kanten aus E_2 nicht die Möglichkeit, die Kante mit einem Knick zu zeichnen, berücksichtigt wird, ist die Modellierung einfach. Wenn allerdings auch diese Optimierungsmöglichkeiten mit berücksichtigt werden sollen, dann erhält man eine Menge von jeweils einfachen, aber in Summe umfangreiche Fallunterscheidungen.

Das Ergebnis in diesem Kapitel sind Mengen von Kanten, die senkrecht, waagrecht oder mit einem statt mit zwei Knicken gezeichnet werden können. Da wir diese Mengen im Rest der Arbeit öfter berücksichtigen müssen, wird für sie noch ein Name vergeben.

Definition 23 (E^s , E^w , E^1)

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Die Menge der Kanten, die in einer planaren, knickoptimalen Zeichnung im Kandinsky-Modell, die das Restriktionensystem schwach respektiert, waagrecht ohne Knick gezeichnet werden, wird mit E^w bezeichnet. Ebenso bezeichnet E^s die Menge der Kanten, die senkrecht gezeichnet werden und E^1 die Menge der Kanten aus E_2 , die mit einem Knick gezeichnet werden. ■

Man beachte den Unterschied zwischen z.B. E_s und E^s . E_s sind die Kanten, die senkrecht gezeichnet werden können, wenn andere Knickeinsparungen ignoriert werden. E^s ist die Auswahl, die in einer knickminimalen Zeichnung verwendet wird.

7.5 Interaktive Knickoptimierung

Im letzten Abschnitt, bei der statischen Knickoptimierung, wird bei einem Berechnungsschritt ignoriert, welche Optimierungen beim letzten Zeichnen des Graphen verwendet wurden. Das soll sich nun ändern. Wir untersuchen, wie die Knickzahl minimiert werden kann, wenn die bisher berechneten Optimierungen bestmöglich beibehalten werden sollen.

Dabei stellen sich zwei grundsätzliche Fragen:

1. In welchem Umfang können die Knickeinsparungen beibehalten werden, wenn eine planare Zeichnung garantiert werden muss? Welche Einsparungen von Knicken vom letzten Zeichenschritt müssen also rückgängig gemacht werden?
2. Wie viele zusätzliche Knicke kann man bei einer solchen lokalen Betrachtungsweise erhalten, wenn vorherige Knickeinsparungen möglichst nicht rückgängig gemacht werden sollen?

Beide Fragen werden wir zunächst beantworten und dann zeigen, wie das Problem algorithmisch gelöst werden kann.

Wir beginnen hier mit der Funktion *InsertEdge*. Einsparungen von Knicken müssen in dem Sinn nur lokal rückgängig gemacht werden, dass nur Kanten betroffen sind, von denen mindestens ein Endknoten in der Fläche liegt, in die die neue Kante eingefügt wird.

Lemma 61

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und die drei Kantenmengen E^s , E^w und E^1 . Es existiere eine planare Zeichnung, die R schwach respektiert und in der alle Kanten aus E^s senkrecht, alle Kanten aus E^w waagerecht und alle Kanten aus E^1 mit einem anstatt mit zwei Knicken gezeichnet werden. Sei außerdem eine restriktionenerhaltende Einfügeoperation $e = \text{InsertEdge}(v_1, v_2, f)$ gegeben. Sei E' die Menge aller Kanten, bei denen mindestens einer der Endknoten in f liegt.

Für den entstehenden Graphen $G = (V, E \cup e)$ existiert eine orthogonale planare Zeichnung im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der zumindest die Kanten $E^s \setminus E'$ senkrecht, die Kanten $E^w \setminus E'$ waagerecht und $E^1 \setminus E'$ mit einem Knick gezeichnet werden.

BEWEIS Betrachten wir zunächst Lemma 21 und 23, in denen gezeigt wird, unter welchen Voraussetzungen eine Kante e' senkrecht bzw. waagerecht ohne Knick gezeichnet werden kann, wenn andere Einsparungen eines Knickes ignoriert werden. Die Voraussetzungen, wann eine Kante waage- oder senkrecht ohne Knick gezeichnet werden kann, ergeben sich immer nur aus den Flächen, in denen die Kanten liegen. Da sich aber durch das Einfügen der Kante nur an den Eigenschaften der Fläche f etwas ändert, können sich die Voraussetzungen für eine Kante nur ändern, wenn sie in f liegt.

Ähnliches gilt bei den Bedingungen, wann eine Kante aus E_2 mit einem Knick gezeichnet werden kann. Die Eigenschaften beziehen sich immer auf Kanten, die zu einem Endknoten der Kante adjazent sind und Flächen, in denen die Endknoten liegen. Dies gilt auch für die Betrachtung der Abhängigkeitsmengen. Da sich Änderungen dieser Bedingungen nur in f ergeben können, gilt die Behauptung auch hier.

Es muss schließlich noch berücksichtigt werden, dass ein Knick bei einer Kante eindeutig einer Kante ohne Knick zugeordnet werden muss. Wir fordern nicht, dass die neue Kante e ohne Knick gezeichnet wird. Deshalb kann es nur passieren, dass an v_1 und v_2 zwei Kanten ohne Knick enden, so dass der Knick bei e nicht mehr einer Kante ohne Knick zugeordnet werden kann. Dazu muss aber die Kante ohne Knick zu v_1 oder v_2 adjazent sein und es liegt wieder einer der Endknoten der Kante in f . ■

Abbildung 7.36(a) zeigt einen Ausschnitt aus der orthogonalen planaren Zeichnung eines Graphen. Es sei $e_1 \in E^s$ und $e_2, e_3 \in E^1$. Wegen der Struktur des Restriktionensystems gilt $e_2, e_3 \in E_{dep}(e_1)$. Wenn die neue Kante e (gestrichelt eingezeichnet) eingefügt wird, dann kann e_1 nicht mehr senkrecht ohne Knick gezeichnet werden. Damit können e_2 und e_3 auch nicht mehr mit einem Knick gezeichnet werden, denn diese Optimierung war davon abhängig, dass e_1 senkrecht ohne Knick gezeichnet wird. Die notwendigen Änderungen werden in Abbildung 7.36(b) gezeigt.

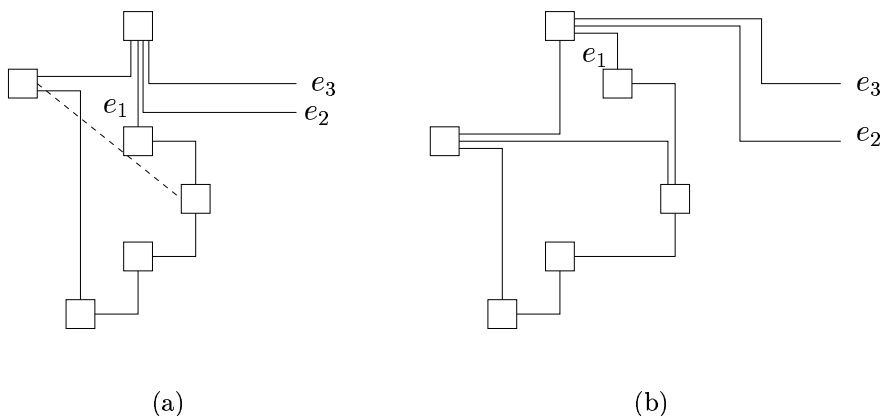


Abbildung 7.36: Mehrere Optimierungen müssen bei einer Einfügeoperation rückgängig gemacht werden, damit eine planare Zeichnung möglich ist.

In Lemma 61 wird die Menge der Kanten, bei denen eine Knickoptimierung rückgängig gemacht werden muss, eingeschränkt. Jedoch kann die Anzahl der Kanten, bei denen die Knickoptimierung rückgängig gemacht werden muss, nicht nach oben abgeschätzt werden, denn die Größe der Abhängigkeitsmengen ist nach oben nicht beschränkt. Es kann aber sein, dass bei allen Kanten einer Abhängigkeitsmenge die Einsparung eines Knickes rückgängig gemacht werden muss.

Das folgende Lemma zeigt die gleichen Überlegungen für die Operation *InsertNode*.

Lemma 62

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und die drei Kantenmengen E^s , E^w und E^1 . Es existiere eine planare Zeichnung, die R schwach respektiert und in der alle Kanten aus E^s senkrecht, alle Kanten aus E^w waagrecht und alle Kanten aus E^1 mit einem anstatt mit zwei Knicken gezeichnet werden. Sei außerdem eine restriktionenerhaltende Einfügeoperation *InsertNode*(e) gegeben, die den Graph G in den Graphen G' überführt. Dabei werde $e = \{v_1, v_2\}$ durch den neuen Knoten v und die beiden neuen Kanten $e_1 = \{v_1, v\}$ und $e_2 = \{v, v_2\}$ ersetzt.

Für den entstehenden Graphen G' existiert eine orthogonale planare Zeichnung im Kandinsky-Modell, die das Restriktionensystem schwach respektiert und in der die Kanten $E^s \setminus e$ senkrecht, die Kanten $E^w \setminus e$ waagrecht und $E^1 \setminus \{e \cup E_{dep}(e)\}$ mit einem Knick gezeichnet werden.

BEWEIS Ob eine Kante senkrecht oder waagrecht ohne Knick gezeichnet werden kann, hängt von lokalen Bedingungen ab, die sich aber durch das Einfügen eines Knotens für keine Kante ändern können (vgl. Lemma 21 und 23).

Auch die Voraussetzungen dafür, ob eine Kante aus E_2 mit einem Knick gezeichnet werden kann, können sich nicht ändern, wenn sie nicht in der Abhängigkeitsmenge von e liegen. ■

Nehmen wir an, dass nach dem Einfügen einer Kante oder eines Knotens so viele Einsparungen eines Knickes erhalten bleiben sollen wie möglich. Dann erhält man im Allgemeinen mehr Knicke, wie wenn man diese Einschränkung nicht hat. Das folgende Beispiel zeigt das. Es sei das Restriktionensystem in Abbildung 7.37(a) gegeben. Die

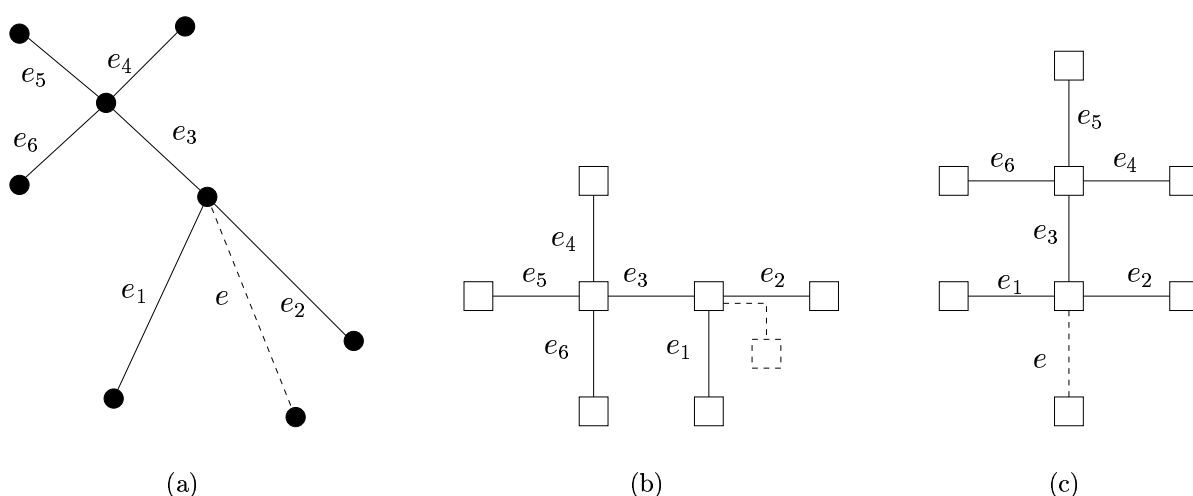


Abbildung 7.37: Beim Einfügen von e müssen Knickoptimierungen aufgehoben werden, damit eine knickminimale Zeichnung möglich ist.

Kante e werde mit der Operation *insertEdge* neu eingefügt. Nehmen wir an, dass vor dem Einfügen $e_1, \dots, e_6 \in E_s$ und $e_1, \dots, e_6 \in E_w$ gilt. Dann ist die Zeichnung, wie sie in Abbildung 7.37(b) dargestellt ist, knickminimal. Hier kann die neue Kante e ohne das Aufheben einer einzigen Optimierung eingefügt werden, allerdings nur mit einem Knick. In Abbildung 7.37(b) ist das gestrichelt eingezeichnet.

Wenn man beliebige Änderungen an den Optimierungen vornehmen darf, dann erhält man als Knickoptimum die Zeichnung in Abbildung 7.37(c). Hier können alle Kanten, auch e , ohne Knick gezeichnet werden.

Das Beispiel zeigt auch, dass es für eine knickoptimale Lösung nötig sein kann, den Verlauf von Kanten zu ändern, deren Endknoten nicht in der Fläche liegen, in der eingefügt wird.²⁶

²⁶ Beachte, dass das Restriktionensystem berücksichtigt werden muss. Deshalb genügt es nicht, einfach nur e_2 senkrecht nach oben zu zeichnen.

Im Folgenden wollen wir einen Algorithmus angeben, der das dynamische Knickoptimierungsproblem löst. Dabei ist der Algorithmus skalierbar. Das heißt, man kann einstellen, bei welchen Kanten Optimierungen rückgängig gemacht bzw. geändert werden dürfen. Abbildung 7.38 zeigt den Algorithmus im Überblick. Die Idee ist dabei, dass für alle Einsparungen eines Knickes, die erhalten bleiben sollen, alle Nachbarn aus G_{opt} entfernt werden.

Führe die Einfügeoperation durch.
Berechne G_{opt} wie bei der statischen Optimierung.
Seien mit den Mengen $E^{s'}$, $E^{w'}$ und $E^{1'}$ die Kanten gegeben, bei denen die Einsparung des Knickes erhalten bleiben soll.
Lösche in G_{opt} alle Nachbarn von Knoten, die einer Kante aus $E^{s'}$, $E^{w'}$ oder $E^{1'}$ entsprechen.
Löse „maximum independent set“ für G_{opt} .

Abbildung 7.38: Algorithmus zur dynamischen Knickoptimierung.

Bei der statischen Optimierung haben wir bei Kanten mit zwei Knicken in den verschiedenen Fällen manche Optimierungsmöglichkeiten verworfen und anstatt dessen eine andere verwendet. Hier kann es durch eine ungeschickte Wahl von $E^{s'}$, $E^{w'}$ und $E^{1'}$ unter Umständen passieren, dass diese Optimierungsvariante nicht zur Verfügung steht. Deshalb müssten wir hier alle möglichen Kombinationen mit E' neu modellieren. Wir sparen uns an dieser Stelle den nötigen Aufwand. In diesem Fall riskiert man als Anwender, dass die Lösung nicht optimal ist.

Wir haben in diesem Kapitel zwei verschiedene Varianten der Knickoptimierung entwickelt, die statische und die dynamische Optimierung. Bei der statischen Optimierung erhalten wir unter den gewählten Rahmenbedingungen eine knickminimale Zeichnung in polynomialer Zeit. Bei der dynamischen Optimierung können die Kantenverläufe bestmöglich erhalten bleiben. Der Preis dafür ist eine höhere Anzahl an Kantenknicken. Allerdings ist die Summe der Kantenknicke unter den in diesem Fall weiter verschärften Rahmenbedingungen wieder optimal. Das verwendete Verfahren liefert einen guten Ansatz für eine Linearzeitheuristik zur Einsparung von Kantenknicken. Denn nur die Auswahl eines „maximum independent set“ benötigt $O(n^2 \log n)$ Laufzeit. Alle anderen Schritte erfolgen in linearer Laufzeit.

Bisher haben wir eine orthogonale Repräsentation des Graphen erzeugt. Im folgenden Kapitel zeigen wir, wie daraus eine Zeichnung gewonnen werden kann. Da in der orthogonalen Repräsentation alle Kantenverläufe festgelegt sind, ist hier das Optimierungskriterium die für die Zeichnung benötigte Fläche.

8 Zeichnen des Graphen

In den beiden letzten Kapiteln haben wir gezeigt, wie aus dem Restriktionensystem eine orthogonale Repräsentation gewonnen werden kann. Dabei gibt es zwei Varianten, nämlich bzgl. einer $<$ -Relation und bzgl. einer \leq -Relation mit einer Knickoptimierung.

Da eine orthogonale Repräsentation eine Standard-Datenstruktur beim orthogonalen Zeichnen ist, können alle bekannten Algorithmen zur Generierung einer Zeichnung aus einer orthogonalen Repräsentation bei diesem Schritt verwendet werden.

Allerdings ist das Problem, eine flächenminimale Zeichnung zu erzeugen, im Allgemeinen NP-vollständig ([Pat99]). Dieses Ergebnis stammt erst aus dem Jahre 1999, obwohl die Fragestellung bereits seit dem ersten Algorithmus von Tamassia ([Tam87]) aus dem Jahre 1987 theoretisch interessant war. Für eine eingeschränkte Klasse von orthogonalen Repräsentationen (engl. „turn-regular“) existiert ein Linearzeitalgorithmus ([BBD⁺99]). Die Einschränkungen, die sich bei unserem Ansatz durch das Restriktionensystem für die möglichen orthogonalen Repräsentationen eines Graphen ergeben, implizieren aber nicht, dass die orthogonale Repräsentation „turn regular“ sein muss. Zusätzlich beruht die erwähnte Arbeit auf dem punktförmigen Knotenmodell. Der Grad eines Knotens ist also auf vier beschränkt. Damit ist dieser Algorithmus für uns nicht anwendbar.

Trotzdem haben auch wir eine eingeschränkte Klasse von orthogonalen Repräsentationen, so dass der allgemeine Beweis der NP-Vollständigkeit aus [Pat99] bei uns nicht anwendbar ist. Damit stellt sich die Frage, ob eine flächenminimale Zeichnung für eine gegebene orthogonale Repräsentation, die dem 4-Sektorenmodell entspricht, polynomial berechenbar ist. Im ersten Abschnitt dieses Kapitels zeigen wir, dass das Zeichnen mit minimaler Fläche auch für Zeichnungen, die dem 4-Sektorenmodell entsprechen, NP-vollständig ist.

8.1 Zeichnen mit minimaler Fläche ist NP-vollständig

Wir zeigen in diesem Abschnitt, dass auch mit den Einschränkungen durch das 4-Sektorenmodell das Erzeugen einer Zeichnung mit minimaler Fläche NP-vollständig ist. Der Beweis verwendet die Idee der Arbeit von [Sto83]. Diese Arbeit beschäftigt sich mit der Einplanung von Elementen auf einem Chip (engl. „floorplaning“). Dabei ist die Anordnung der Bauelemente bereits gegeben. Jedes Bauelement wird durch ein Rechteck repräsentiert und es muss nun festgelegt werden, wie das Element auf dem Chip ausgerichtet werden soll (d. h., es muss hochkant oder quer platziert werden). In der Arbeit von Stockmeyer wird bewiesen, dass dieses Problem NP-vollständig ist. Wir werden nun un-

ser Problem von 3-Partition reduzieren. Dazu folgen zunächst die formalen Definitionen der Probleme. Wir nennen unser Problem, abgeleitet von „4-Sektorenmodell, minimale Fläche“, *4SMMF*.

Definition 24 (4SMMF)

Instanz: Ein Graph G mit einem legalen Restriktionensystem R , eine orthogonale Repräsentation von G , die R schwach respektiert und eine natürliche Zahl K .

Frage: Existiert eine, der orthogonalen Repräsentation von G entsprechende, orthogonale, planare Zeichnung mit Fläche $\leq K$? ■

Wir übernehmen die Definition für 3-Partition aus [GJ83].

Definition 25 (3-Partition)

Instanz: Eine Menge A mit $3m$ Elementen, eine Zahl $B \in \mathbb{Z}$, so dass $\forall a \in A : s(a) \in \mathbb{Z}$ mit $B/4 < s(a) < B/2$ und $\sum_{a \in A} s(a) = mB$.

Frage: Kann A in m Mengen A_1, \dots, A_m partitioniert werden, so dass: $\sum_{a \in A_i} s(a) = B$? ■

Die Bedingung $B/4 < s(a) < B/2$ stellt sicher, dass in jedem A_i genau drei Elemente liegen müssen.

Wir zeigen, dass 4SMMF NP-vollständig ist. Die Tatsache, dass 4SMMF in NP lösbar ist, ist eine unmittelbare Folgerung aus der Tatsache, dass flächenminimales Zeichnen für beliebige orthogonale Repräsentationen in NP lösbar ist (vgl. [Pat99]). Damit bleibt zu zeigen, dass 4SMMF NP-schwer ist.

Wir reduzieren 3-Partition auf 4SMMF. Der Beweis ist auch für das punktförmige Knotenmodell gültig, da bei der Konstruktion des Graphen darauf geachtet wird, dass auf jeder Seite eines Knotens nur eine Kante enden kann. Das heißt, unser Beweis ist auch dann gültig, wenn der Grad eines Knotens durch vier beschränkt ist.

Gegeben sei eine Instanz von 3-Partition. Dann konstruieren wir in einem ersten Schritt die orthogonale Repräsentation eines Graphen, wie sie in Abbildung 8.1 dargestellt ist.

Die orthogonale Repräsentation besteht in jeder Zeile aus $3 \cdot m$ Flächen, die jeweils durch einen Teilgraphen in Form eines Rechteckes begrenzt sind. Dabei gibt es abwechselnd eine Zeile, in der die Rechtecke mit $X_{i,j}$ und eine Zeile, in der die Rechtecke mit $Y_{i,j}$ beschriftet sind. Es gibt m Zeilen, in denen die Rechtecke mit $X_{i,j}$ beschriftet sind und $m - 1$ Zeilen, in denen die Rechtecke mit $Y_{i,j}$ beschriftet sind. Dabei sind die Rechtecke nicht senkrecht untereinander angeordnet, sondern stehen „auf Lücke“. Zum Ausgleich der verschiedenen Höhen sind an den Ecken der bisher beschriebenen Rechtecke kleine Rechtecke angeordnet. Sie sind nicht nummeriert, denn ihre einzige Funktion ist es, die Höhenunterschiede auszugleichen, die dadurch entstehen, dass die Rechtecke nicht direkt übereinander liegen.

Man beachte, dass diese orthogonale Repräsentation mit dem in Kapitel 5 beschriebenen Algorithmus und einer nachfolgenden Knickoptimierung erzeugt werden kann. Dazu muss der Graph mit dem Restriktionensystem aus Abbildung 8.2 generiert werden. Die

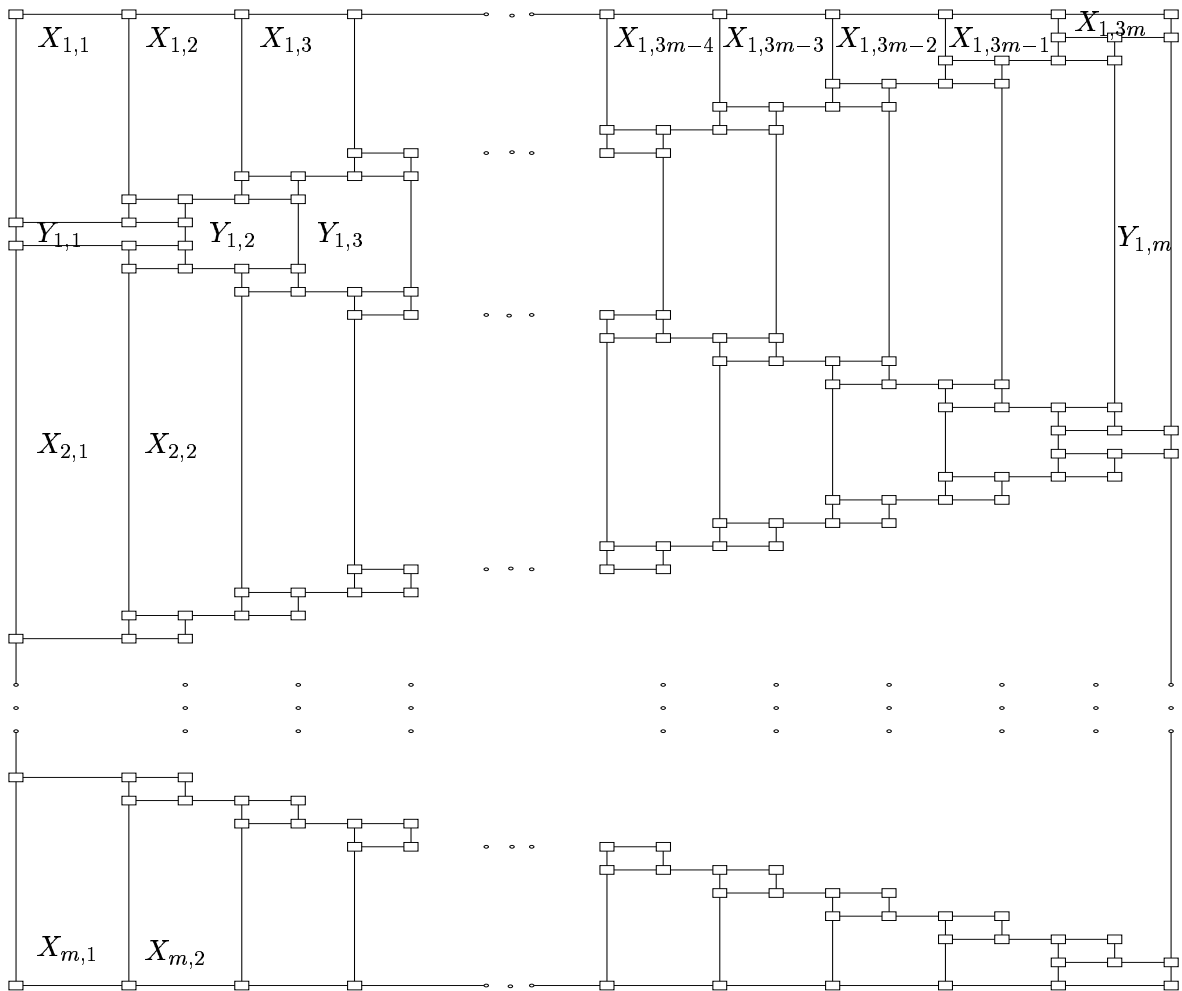


Abbildung 8.1: Erster Schritt zur Konstruktion der orthogonalen Repräsentation zum Beweis, dass die Flächenoptimierung NP-vollständig ist.

orthogonale Repräsentation aus Abbildung 8.1 entspricht diesem Restriktionensystem und ist knickfrei. Damit ist die orthogonale Repräsentation eine mögliche Lösung der Knickminimierung für das Restriktionensystem.

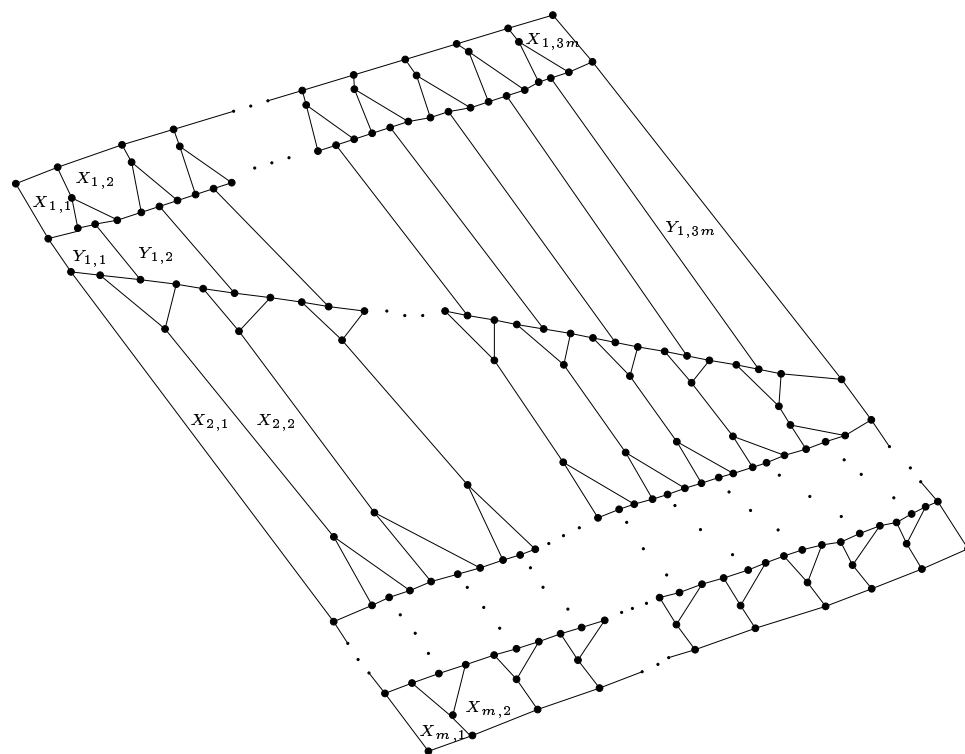


Abbildung 8.2: Restriktionensystem, das der orthogonalen Repräsentation in Abbildung 8.1 entspricht.

Für die weitere Konstruktion benötigen wir als Hilfsstruktur die orthogonale Repräsentation aus Abbildung 8.3(a).

Diese orthogonale Repräsentation hat die wichtige Eigenschaft, dass zwei Zeichnungen mit minimaler Fläche existieren, die in den Abbildungen 8.3(a) und 8.3(b) dargestellt sind: Einmal mit Fläche 5×4 (Abb. 8.3(a)) und einmal mit Fläche 4×5 (Abb. 8.3(b)). Wir bezeichnen diese orthogonale Repräsentation im Folgenden mit M .

In dem im ersten Schritt generierten Graphen müssen die Felder $X_{i,j}$ und $Y_{i,j}$ gefüllt werden. Dabei gehen wir nach folgender Idee vor: In den Feldern einer Spalte $X_{i,j}$ ($1 \leq i \leq m$) wird das Gewicht $s(a_j)$ kodiert. In einer Zeile mit den Feldern $X_{i,j}$ ($1 \leq j \leq 3m$) wird die i -te Partition dargestellt. Dabei wird jeweils der Teilgraph, der sich in Feld $X_{i,j}$ befindet, „hoch und schmal“ gezeichnet (vgl. Abbildung 8.3(b)), wenn a_j in der Partition i liegt. Wenn a_j nicht in der Partition i liegt, dann wird der Teilgraph „flach und breit“ gezeichnet (vgl. Abbildung 8.3(a)). Im Rest des Kapitels verwenden wir die Bezeichnung $X_{i,j}$ auch für den Subgraphen in Feld $X_{i,j}$.

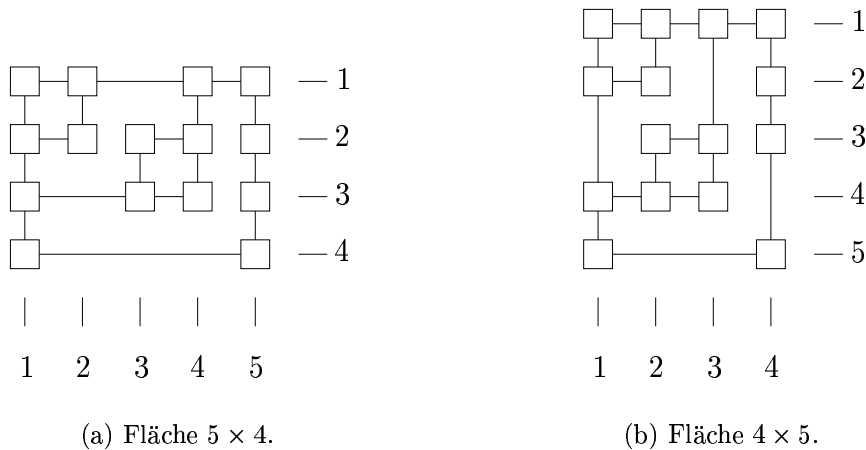


Abbildung 8.3: Hilfsgraph mit zwei verschiedenen, flächenoptimalen Zeichnungen.

Für alle $i, 1 \leq i \leq m$ und alle $j, 1 \leq j \leq 3m$ wird das Feld $X_{i,j}$ mit einer orthogonalen Repräsentation wie in Abbildung 8.4 gefüllt. In der Zeichnung ist der vorherige Teilgraph um Feld $X_{i,j}$ zur Verdeutlichung fett gezeichnet.¹

Die grau unterlegten Knoten im Rahmen der Box kommen neu zum Graphen hinzu. In jedem Feld $X_{i,j}$ werden $s(a_j)$ Kopien von der orthogonalen Repräsentation M erzeugt, wobei a_j das j -te Element der Menge A aus der 3-Partition ist. Je zwei Kopien werden durch ein Verschmelzen der rechten Seite und der linken Seite verbunden. Damit erhalten wir einen rechteckigen Subgraphen, der alle Kopien von M enthält. Damit werden $s(a_j)$ Spalten in der Breite gespart, wenn der Teilgraph „hoch und schmal“ gezeichnet wird.

Dieser Subgraph wird dann an die Box um Feld $X_{i,j}$ angeschlossen. Dazu werden die Eckpunkte des Subgraphen mit den neuen Knoten im Rahmen um die Box verbunden. Dabei bestehen die beiden linken Verbindungen jeweils aus einer Kette der Länge l . Die Berechnung von l geben wir an, nachdem die Subgraphen $Y_{i,j}$ erklärt sind. Die Subgraphen $Y_{i,j}$ haben verschiedene Höhen und die Ketten werden zum Ausgleich dieser verschiedenen Höhen verwendet. Die beiden rechten Verbindungen bestehen aus je einer Kante.

Auch mit dieser Erweiterung entspricht die orthogonale Repräsentation dem 4-Sektorenmodell. Abbildung 8.5 zeigt ein entsprechendes Restriktionensystem. Ebenso wie für den Graphen in Abbildung 8.2 existiert eine Sequenz von Einfügeoperationen, so dass das Restriktionensystem mit dem Algorithmus in Kapitel 5 erzeugt werden kann. In der Abbildung ist die Box um den Subgraphen $X_{i,j}$ zur Verdeutlichung wieder fett dargestellt.

¹ Die Knoten zwischen den Ecken der Box an der oberen, unteren und rechten Seite sind je nach Lage im Gesamtgraphen nicht in jedem Feld $X_{i,j}$ vorhanden. Es gibt z. B. im Feld $X_{1,1}$ keine Knoten an der Oberseite und im Feld $X_{1,3m}$ keine an der rechten Seite.

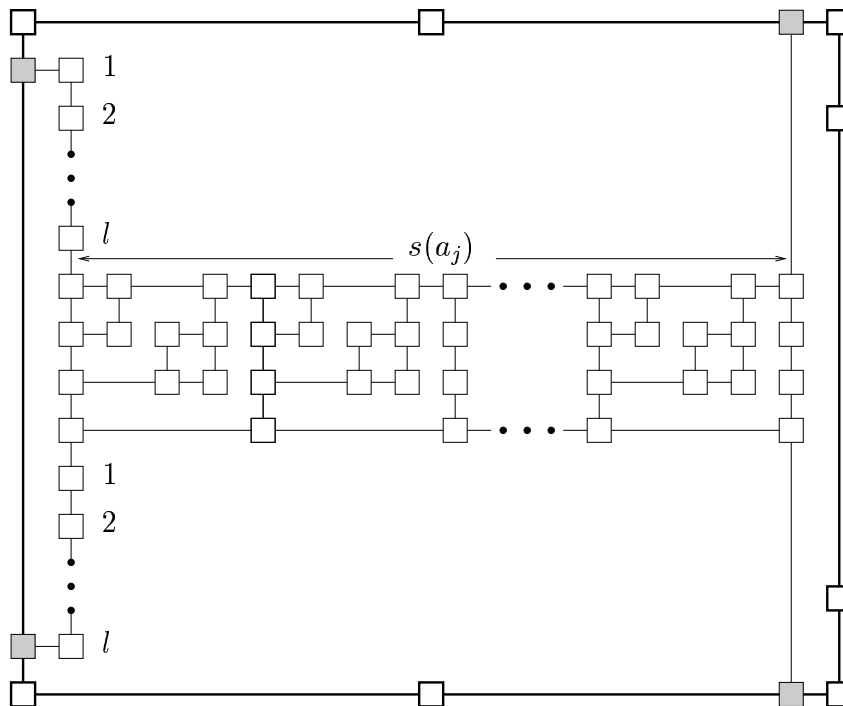


Abbildung 8.4: Orthogonale Repräsentation des Graphen $X_{i,j}$.

Außerdem müssen noch die Felder $Y_{i,j}$ gefüllt werden. Diese Felder erfüllen folgenden Zweck: Über die Höhe des Gesamtgraphen² wird kontrolliert, dass für alle $i, 1 \leq i \leq 3m$ nur einer der Subgraphen $X_{i,j}$ „hoch“ gezeichnet wird. Die Felder $Y_{i,j}$ dienen dann jeweils als Zwischenstück, das sicherstellt, dass sich zwei- oder mehrmaliges „hochkantiges“ Zeichnen von Subgraphen $X_{i,j}, 1 \leq i \leq m$ in der Gesamthöhe auswirkt.

Die Höhe eines Pfades P mit maximaler Höhe durch Felder $X_{i,j}$ ($1 \leq i \leq m$) und $Y_{i,j}$ ($1 \leq i \leq m - 1$) setzt sich für ein festes j aus den Höhen der $X_{i,j}$ und der $Y_{i,j}$ zusammen³. Durch die Konstruktion muss jetzt dafür gesorgt werden, dass jeder Pfad, der nicht nur durch diese Felder, sondern zusätzlich auch durch ein Feld $X_{o,j}$ oder $Y_{o,j}$ mit $o \neq i$ verläuft, geringere Höhe hat als P . Das wird dadurch erreicht⁴, dass die Höhe der $Y_{i,j}$ von rechts nach links immer größer wird. Wir füllen jedes Feld $Y_{i,j}$ mit einer Kette der Länge $j \cdot 4$ (Abbildung 8.6). Die Kette wird an den Knoten auf der oberen und unteren Kante um Feld $Y_{i,j}$ aufgehängt. Falls dieser Knoten nicht existiert⁵, dann wird die obere und untere Kante jeweils aufgetrennt und ein neuer Knoten eingefügt.

Um die verschiedenen Höhen der $Y_{i,j}$ auszugleichen, müssen die Ketten der Länge l in den Feldern $X_{i,j}$ angepasst werden. Dazu legen wir l wie folgt fest:

² Und damit auch für jeden senkrechten Pfad.

³ Plus eine additive Konstante für die Konstruktion der Boxen.

⁴ Der Beweis folgt nach der Konstruktion.

⁵ Das ist bei den Feldern $Y_{i,3m}, 1 \leq i \leq m - 1$ der Fall.

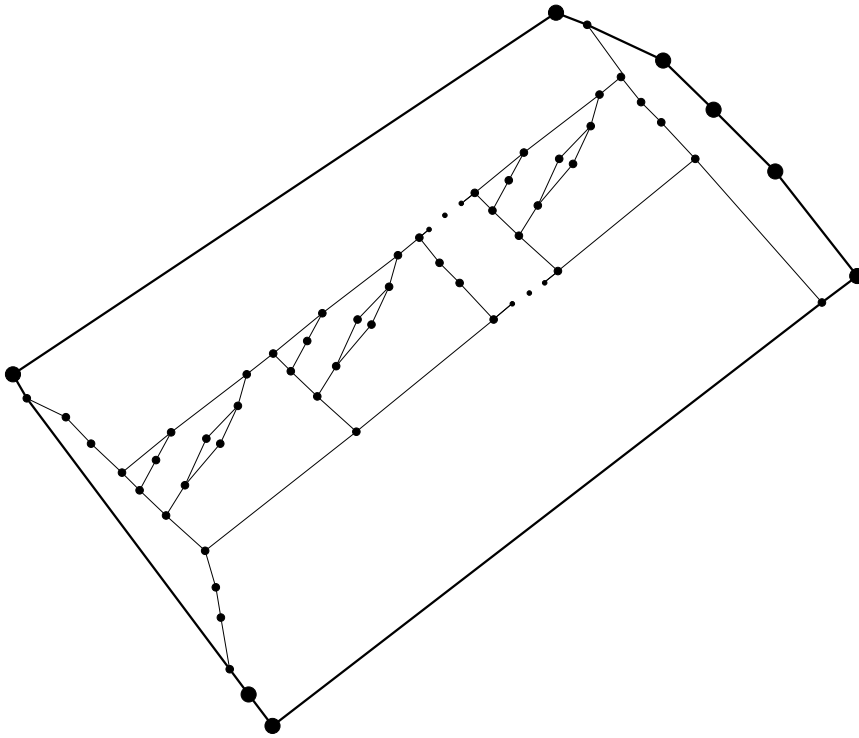


Abbildung 8.5: Restriktionensystem zur orthogonalen Repräsentation in Abbildung 8.4.

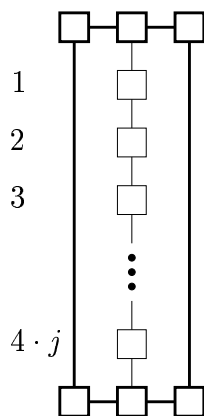


Abbildung 8.6: Orthogonale Repräsentation des Graphen $Y_{i,j}$.

$$l = \begin{cases} (3m - j) \cdot 1 & \text{falls } i \in \{1, m\} \\ (3m - j) \cdot 2 & \text{falls } 2 \leq i \leq m - 1 \end{cases}$$

Man beachte, dass in jedem $X_{i,j}$ zwei Pfade mit dieser Länge vorkommen, einmal oberhalb der Kopien von Graph M und einmal unterhalb. In Summe benötigt der Subgraph in Feld $X_{i,j}$ also zweimal die Länge l plus die Höhe der Kopien von M .

Den bisher konstruierten Graphen benötigen wir im Folgenden mehrfach und bezeichnen ihn deshalb mit U . In einem letzten Schritt wird der Graph U noch einmal, wie in Abbildung 8.7 dargestellt, erweitert. Es werden noch Ketten von Knoten, wie in der Abbildung gezeigt, eingefügt. Der Grund dafür ist einfach. Wir wollen zeigen, dass das Zeichnen mit minimaler Fläche NP-schwer ist. Für den Beweis benötigen wir eine Zeichnung von Graph U mit einer bestimmten maximalen Breite und Höhe. Es kann aber eine weitere Zeichnung mit noch kleinerer Fläche geben, wenn z. B. die Breite größer ist als die von uns gewünschte maximale und dafür aber die Höhe viel kleiner ist. Mit der zusätzlichen Erweiterung von U erreichen wir, dass das nicht mehr möglich ist.⁶

Den neu konstruierten Graphen bezeichnen wir mit V . Dabei sind die Werte W und H wie folgt definiert:

- $W = 4Bm + 6m - B + 1$
- $H = 12m^2 - 6m + 1$.

Auch für die vollständige orthogonale Repräsentation V gibt es einen Graphen mit einem Restriktionensystem, das von V schwach respektiert wird. In Abbildung 8.8 ist das entsprechende Restriktionensystem dargestellt. Es ist eine Erweiterung des Restriktionensystems aus Abbildung 8.2.

Damit ist die Konstruktion des Graphen vollständig. Es ist noch zu zeigen, dass die orthogonale Repräsentation genau dann eine Zeichnung mit Fläche $\leq K$ hat⁷, wenn die zugehörige Instanz von 3-Partition eine Lösung hat. Um den Beweis von Satz 14 kürzer zu halten, beweisen wir zunächst eine Reihe von Hilfsbehauptungen. Die ersten beiden beschäftigen sich damit, wie lange ein Pfad von oben nach unten durch den Graphen, bis zu einem bestimmten Punkt, sein kann.

Behauptung 1

Sei $1 \leq i \leq m - 1$, $1 \leq j \leq 3m$. Sei P ein beliebiger Pfad durch die Graphen $X_{o,j}$, $1 \leq o \leq i$ und $Y_{o,j}$, $1 \leq o \leq i - 1$ oder die Kanten der umgebenden Felder. Gelte, dass maximal eines der $X_{o,j}$, $1 \leq o \leq i$ hoch gezeichnet⁸ ist.

Für die Länge l_p des Pfades P vom oberen Ende des Graphen U bis zu den Knoten am unteren Ende der Box $X_{i,j}$ gilt

$$12mi + 6i - 6m - 2j + 1 \geq l_p \geq 12mi + 6i - 6m - 2j.$$

⁶ Der Beweis folgt nach der Konstruktion des Graphen.

⁷ K muss dazu noch definiert werden.

⁸ Das heißt, die Kopien von Graph M in Feld $X_{i,j}$ werden hoch und schmal gezeichnet.

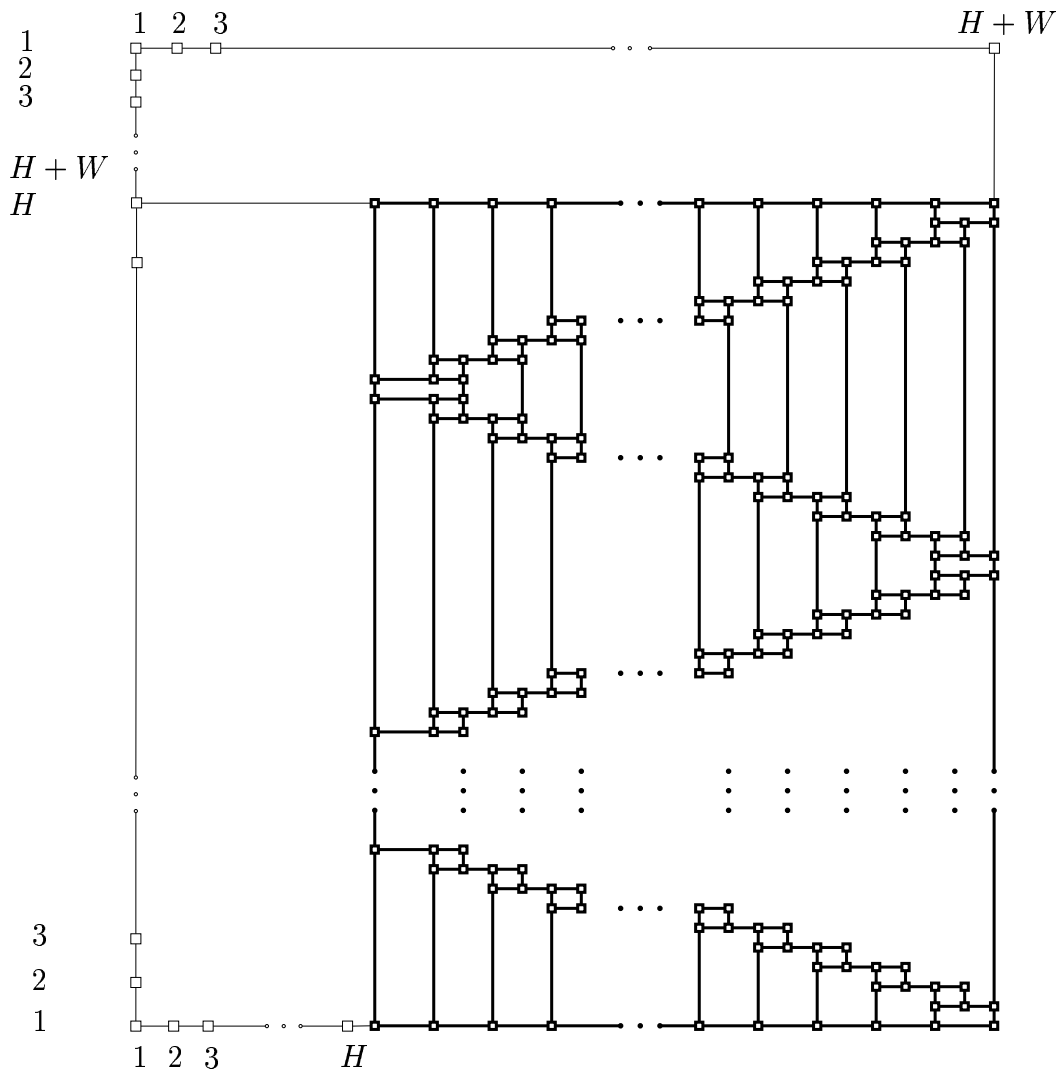


Abbildung 8.7: Endgültiger Graph V .

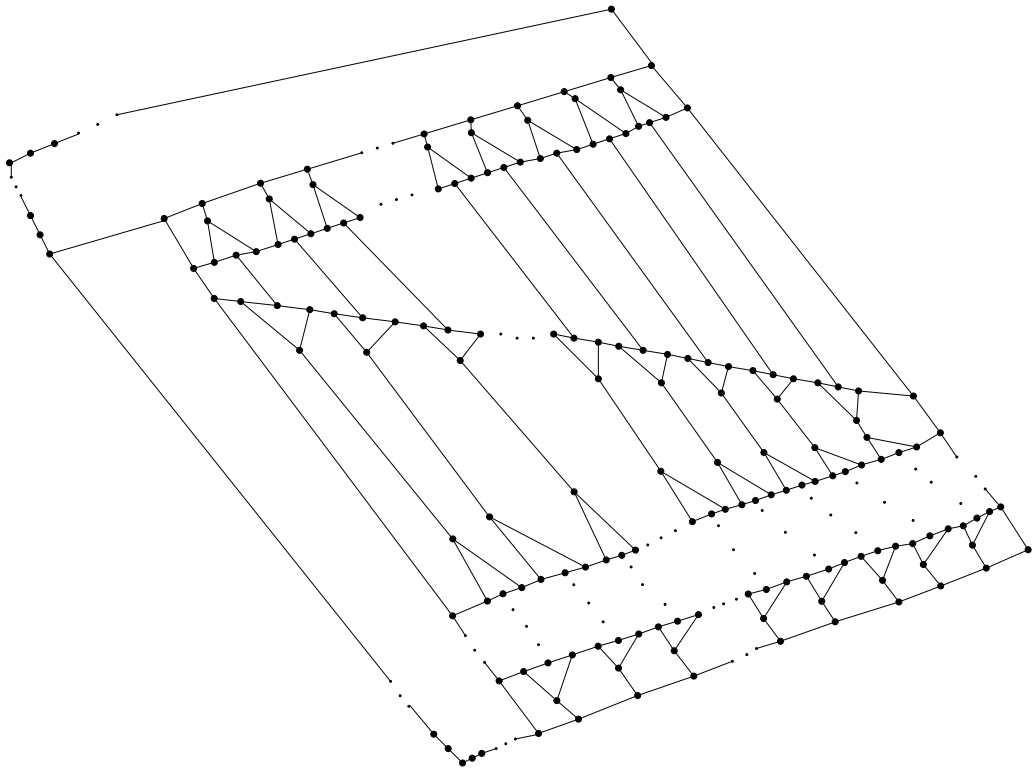


Abbildung 8.8: Restriktionensystem von Graph V .

BEWEIS Wir zeigen zunächst $l_p \geq 12mi + 6i - 6m - 2j$. Dazu berechnen wir die minimale Höhe, die wir erreichen können. Das erreichen wir, wenn alle Felder $X_{i,j}$ flach gezeichnet werden.

$$\begin{aligned}
 l_p &\geq 2i+ && \text{(Knoten im Rahmen von } U) \\
 &4j \cdot (i-1)+ && \text{(Höhe der } Y_{o,j}) \\
 &(3m-j) \cdot 2+ && \text{(Ketten in } X_{1,j}) \\
 &(3m-j) \cdot 4 \cdot (i-1)+ && \text{(Ketten in } X_{2,j} \text{ bis } X_{i,j}) \\
 &4i && \text{(alle } X_{o,j} \text{ flach gezeichnet)} \\
 &= 6i + 4ji - 4j + 6m - 2j+ \\
 &12mi - 4ji - 12m + 4j \\
 &= 12mi + 6i - 6m - 2j
 \end{aligned}$$

Außerdem ist $12mi + 6i - 6m - 2j + 1 \geq l_p$ zu zeigen. Dazu muss die maximale Höhe des Pfades berechnet werden. Das erreichen wir, wenn ein Feld $X_{k,j}$ hoch gezeichnet wird. Mehr ist nicht erlaubt. Wenn $X_{k,j}$ hoch gezeichnet wird, dann benötigt es eine um eins größere Höhe, wie wenn es flach gezeichnet wird. Die Höhe des Pfades ist in diesem Fall also um eines höher als die eben berechnete minimale Höhe. Damit folgt die zweite Ungleichung für l_p aus der ersten, da nur ein $X_{o,j}$ hoch gezeichnet sein darf. ■

Behauptung 2

Sei $1 \leq i \leq m-1$, $1 \leq j \leq 3m$. Sei P ein beliebiger Pfad durch die Graphen $X_{o,j}$, $1 \leq o \leq i$ und $Y_{o,j}$, $1 \leq o \leq i$ oder die Kanten der umgebenden Felder. Gelte, dass maximal eines der $X_{o,j}$, $1 \leq o \leq i$ hoch gezeichnet ist.

Für die Länge l_p des Pfades P vom oberen Ende des Graphen U bis zu den Knoten am unteren Ende der Box $Y_{i,j}$ gilt

$$12mi + 6i - 6m + 2j \geq l_p \geq 12mi + 6i - 6m + 2j - 1.$$

BEWEIS Wir zeigen $l_p \geq 12mi + 6i - 6m + 2j - 1$. Dazu berechnen wir die minimale Höhe, die erreicht wird wenn alle Felder $X_{i,j}$ flach gezeichnet werden.

$$\begin{aligned}
 l_p &\geq 2(i-1) + 1+ && \text{(Knoten im Rahmen von } U) \\
 &4ji+ && \text{(Höhe der } Y_{o,j}) \\
 &(3m-j) \cdot 2+ && \text{(Ketten in } X_{1,j}) \\
 &(3m-j) \cdot 4 \cdot (i-1)+ && \text{(Ketten in } X_{2,j} \text{ bis } X_{i,j}) \\
 &4i && \text{(alle } X_{o,j} \text{ quer gezeichnet)} \\
 &= 2i - 1 + 4ij + 6m - 2j+ \\
 &12mi - 4ji - 12m + 4j + 4i \\
 &= 12mi + 6i - 6m + 2j - 1
 \end{aligned}$$

Außerdem ist $12mi + 6i - 6m + 2j \geq l_p$ zu zeigen. Dazu muss die maximale Höhe des Pfades berechnet werden. Das erreichen wir, wenn ein Feld $X_{k,j}$ hoch gezeichnet wird, da mehr nicht hoch gezeichnet werden dürfen. Wenn $X_{k,j}$ hoch gezeichnet wird, dann benötigt es eine um eins größere Höhe. Damit folgt die zweite Ungleichung für l_p aus der ersten, da nur ein $X_{o,j}$ hoch gezeichnet sein darf. ■

Die folgende Behauptung zeigt, dass der Graph V nur dann mit Fläche $\leq K$ gezeichnet werden kann, wenn der Graph U mit Breite $\leq W$ und Höhe $\leq H$ gezeichnet werden kann. Wir werden dieses Lemma nutzen, um uns bei der Reduktion von 3-Partition nur noch mit der Breite und Höhe von U zu beschäftigen.

Behauptung 3

Sei $K = (H + W) \cdot (2H + W)$. Der Graph V kann mit einer Fläche $\leq K$ gezeichnet werden \Leftrightarrow der Graph U kann mit Breite $\leq W$ und Höhe $\leq H$ gezeichnet werden.

BEWEIS Mit Breite von Graph $U = W'$ und Höhe von $U = H'$ gilt:

Die Breite vom Gesamtgraphen V ist mindestens $H + W$ durch die Kette am oberen Ende des Graphen. Außerdem ist die Breite mindestens $H + W'$ durch die Kette der Länge H am unteren Ende des Graphen, wobei danach der Graph U mit Breite W' gezeichnet werden muss. Die Breite vom Gesamtgraphen V ist deshalb $\max\{H + W, H + W'\} = H + \max\{W, W'\}$.

Entsprechend ergibt sich, dass die Höhe von Graph $V = H + W + \max\{H, H'\}$ ist.

Damit ist die Gesamtfläche $(H + \max\{W, W'\}) \cdot (H + W + \max\{H, H'\})$. Falls also $W' > W$ oder $H' > H$ gilt, dann kann der Graph V nicht mit Fläche $\leq K$ gezeichnet werden. ■

Im Beweis zum folgenden Satz verwenden wir Abbildung 8.9 zur Veranschaulichung. In der Abbildung ist ein Ausschnitt von U dargestellt. Dabei ist die Rahmenkonstruktion fett gezeichnet. Die Subgraphen $X_{i,j}$ und $Y_{i,j}$ und ihre Verbindungsstruktur werden durch die kleinen Rechtecke symbolisiert. Die Subgraphen $X_{i,j}$ und $Y_{i,j}$ sind nicht vollständig dargestellt.

Satz 14

Eine Instanz I von 3-Partition hat eine Lösung \Leftrightarrow Der zu I konstruierte Graph V kann mit Fläche $K \leq (H + W) \cdot (2H + W)$ gezeichnet werden.

Damit ist 4SMMF NP-vollständig, da 3-Partition nach 4SMMF reduziert werden kann.

BEWEIS Es genügt nach Behauptung 3 zu zeigen, dass 3-Partition genau dann eine Lösung hat, wenn der Teilgraph U vom zugehörigen Graphen V Breite $\leq W$ und Höhe $\leq H$ hat.

\Leftarrow : Gegeben sei eine Zeichnung von U mit Höhe $\leq H$ und Breite $\leq W$. Dann gilt für alle $i \in \{1, \dots, m\}$:

(1) $\exists j_1, \dots, j_o : \sum_{k=1}^o s(a_{j_k}) \geq B$ und $X_{i,j_k}, 1 \leq k \leq o$ werden hoch und schmal gezeichnet.

Wir zeigen (1). Angenommen, es gibt ein i , so dass (1) nicht erfüllt ist. Wir zeigen, dass dann die Breite von U größer als W ist. Seien j_1, \dots, j_o so, dass $X_{i,j_k}, 1 \leq k \leq o$ hoch und schmal gezeichnet werden. Da (1) nach Annahme nicht gilt, gilt $\sum_{k=1}^o s(a_{j_k}) < B$. Die übrigen Felder werden breit und flach gezeichnet.

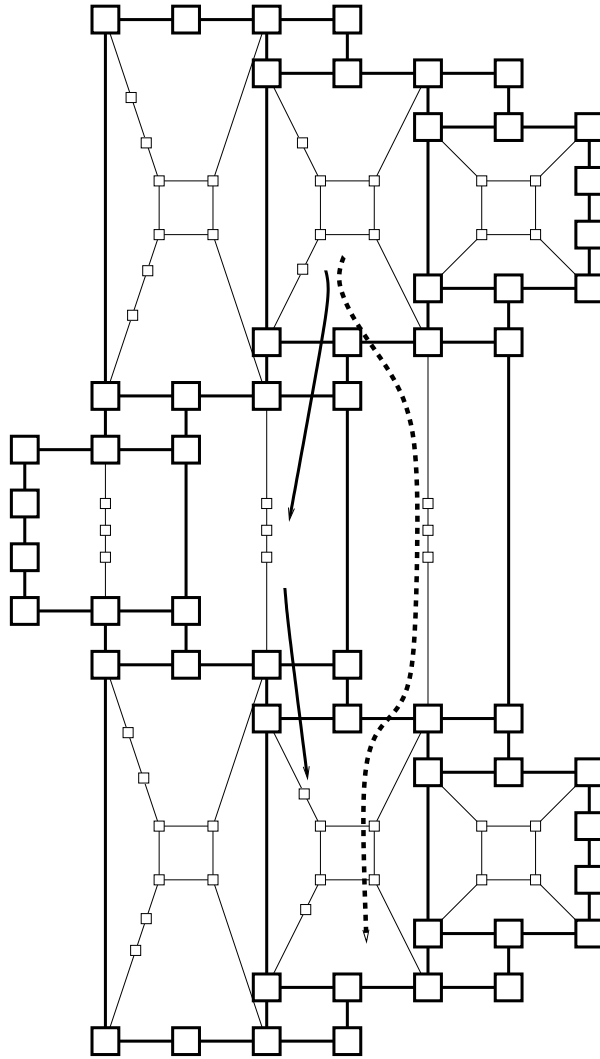


Abbildung 8.9: Graph zum Beweis von Satz 14.

Sei M die Menge der j , so dass die Felder $X_{i,j}$ breit gezeichnet werden. Da (1) nach Annahme nicht gilt, folgt für M :

$$\sum_{j \in \{1, \dots, 3m\} \setminus M} s(a_j) < B \text{ und } \sum_{j \in M} s(a_j) > Bm - B$$

und, da $\sum_{a_j \in A} s(a_j) = Bm$, existiert ein $c \in \mathbb{N}^+$:

$$(*) \quad \sum_{j \in \{1, \dots, 3m\} \setminus M} s(a_j) = B - c \wedge \sum_{j \in M} s(a_j) = Bm - B + c.$$

Wir wissen, dass jedes breit gezeichnete $X_{i,j}$ die Breite $4 \cdot s(a_i) + 1$ und jedes hoch gezeichnete $X_{i,j}$ die Breite $3 \cdot s(a_i) + 1$ benötigt. Damit benötigt die Zeile i folgende Gesamtbreite:

$$\begin{aligned} W' &= 3m + 1 + && \text{(Knoten im Rahmen um } U) \\ &4 \cdot \sum_{j \in M} s(a_j) + |M| + && \text{(breit gezeichnete } X_{i,j}) \\ &3 \cdot \sum_{j \in \{1, \dots, 3m\} \setminus M} s(a_j) + 3m - |M| && \text{(schmal gezeichnete } X_{i,j}) \\ &= 3m + 1 + \\ &4 \cdot (Bm - B + c) + |M| + && \text{(siehe (*))} \\ &3 \cdot (B - c) + 3m - |M| && \text{(siehe (*))} \\ &= 3m + 1 + 4Bm - 4B + 4c + 3B - 3c + |A| \\ &= 4BM + 6m - B + 1 + c \end{aligned}$$

Es wäre also $W' > W$, was ein Widerspruch zur Annahme ist. Es muss also (1) gelten.

Wegen der Forderung $B/4 \leq s(a_j) \leq B/2$ gilt in (1) zusätzlich: $o \geq 3$. In jeder Zeile gibt es also mindestens drei Felder, so dass die Kopien von Graph M im Feld hoch und schmal gezeichnet werden.

Außerdem gilt für alle $j \in \{1, \dots, 3m\}$, also für jede Spalte des Graphen:

(2) maximal eines der Felder $X_{i,j}$, $1 \leq i \leq m$ ist hoch gezeichnet.

Wir zeigen (2). Angenommen, es existiert ein j , so dass (2) nicht erfüllt ist. Wir betrachten den vertikalen Pfad durch die $X_{i,j}$, $1 \leq i \leq m$ und $Y_{i,j}$, $1 \leq i \leq m - 1$ und berechnen die Höhe H' von U . Seien also mindestens zwei der $X_{i,j}$, $1 \leq i \leq m$ hoch gezeichnet. Wir wissen, dass jedes hoch gezeichnete M Höhe fünf hat und jedes breit gezeichnete Höhe vier. Damit gilt folgende Ungleichung für H' :

$$\begin{aligned} H' &\geq 2m + && \text{(Knoten der Rahmenkonstruktion von } U) \\ &4j \cdot (m - 1) + && \text{(Höhe der } Y_{i,j}) \\ &(3m - j) \cdot 2 \cdot 2 + && \text{(Ketten in } X_{1,j} \text{ und } X_{m,j}) \\ &(3m - j) \cdot 4 \cdot (m - 2) + && \text{(Ketten in } X_{2,j} \text{ bis } X_{m-1,j}) \\ &4 \cdot (m - 2) + && \text{(quer gezeichnete } X_{i,j}) \\ &5 \cdot (2) && \text{(hoch gezeichnete } X_{i,j}) \\ &= 2m + 4jm - 4j + 12m - 4j + \\ &12m^2 - 24m - 4jm + 8j + 4m + 2 \\ &= 12m^2 - 6m + 2 \end{aligned}$$

Damit gilt $H' > H$ und die Behauptung (2) muss gelten.

Damit haben wir eine Lösung von 3-Partition, in der a_j genau dann in Partition i liegt, wenn Feld $X_{i,j}$ hoch gezeichnet ist.

Da für jedes j nur eines der Felder $X_{i,j}$ hoch gezeichnet ist (Behauptung (2)), kommt jedes a_j nur in einer Partition vor.

Wegen (1) sind für jedes i mindestens drei $X_{i,j}$ hoch gezeichnet. Wegen (2) sind aber maximal $3m$ $X_{i,j}$ hoch gezeichnet und somit sind für jedes i genau drei $X_{i,j}$ hoch gezeichnet. In jeder Partition liegen also drei $a \in A$ und es gilt wegen (2) $\forall i \in m : \sum_{a \in A_i} s(a) \geq B$. Da aber $\sum_{a \in A} s(a) = mB$ und m Partitionen existieren, muss $\forall i \in m : \sum_{a \in A_i} s(a) = B$ gelten.

\Rightarrow : Gegeben sei eine Lösung von 3-Partition. Dann konstruieren wir die Zeichnung für den Graphen wie folgt: Falls $a_j \in A_i$: Zeichne die Kopien von Graph M in Feld $X_{i,j}$ wie in Abbildung 8.3(b) hoch und schmal. Andernfalls zeichne die Kopien von M wie in Abbildung 8.3(a) breit und niedrig.

Dann hat U eine Breite $\leq W$ und eine Höhe $\leq H$. Wir zeigen zunächst, dass die Breite kleiner gleich W ist. Dazu muss jeder Pfad quer durch U mit Breite $\leq W$ gezeichnet werden können. Hierbei sind Pfade, die ein $Y_{i,j}$ enthalten, unkritisch. Denn ihre Breite ist klein im Vergleich zur Breite der $X_{i,j}$ und damit können die Ränder ihrer Felder beliebig zwischen den Eckpunkten der $X_{i,j}$ platziert werden.

Für alle $i, 1 \leq i \leq m$ gilt für die Länge L eines Pfades durch die $X_{i,j}, 1 \leq j \leq 3m$ (mit M wieder die Menge der j , so dass die Felder $X_{i,j}$ breit gezeichnet werden):

$$\begin{aligned}
 L &\leq 3m + 1 + && \text{(Knoten der Rahmenkonstruktion von } U) \\
 &4 \cdot \sum_{j \in M} s(a_j) + |M| + && \text{(breit gezeichnete } X_{i,j}) \\
 &3 \cdot \sum_{j \in \{1, \dots, 3m\} \setminus M} s(a_j) + 3m - |M| && \text{(schmal gezeichnete } X_{i,j}) \\
 = &3m + 1 + && \\
 &4 \cdot (Bm - B) + |M| + && \left(\sum_{j \in M} s(a_j) = Bm - B \right) \\
 &3 \cdot B + |A \setminus M| && \left(\sum_{a_j \in \{1, \dots, 3m\} \setminus M} s(a_j) = B \right) \\
 = &3m + 1 + 4Bm - 4B + 3B + |A| \\
 = &4BM + 6m - B + 1
 \end{aligned}$$

Also ist der Graph mit Breite $\leq W$ darstellbar.

Weiter ist zu zeigen, dass der Graph mit Höhe $\leq H$ gezeichnet werden kann. Wir betrachten zunächst nur alle vertikalen Pfade durch den Graphen für die gilt:

(3) Es existiert ein $j, 1 \leq j \leq 3m$ so dass alle Kanten in einem $X_{i,j}, 1 \leq i \leq m$ oder $Y_{i,j}, 1 \leq i \leq m - 1$ oder in einem der diese Subgraphen umgebenden Felder $X_{i,j}$ liegen.

In Abbildung 8.9, das einen Ausschnitt aus U zeigt, ist das durch den gestrichelten Pfeil dargestellt. Wir berechnen die maximale Länge L' dieser Pfade:

$$\begin{aligned}
 L' &\leq 2m+ && \text{(Knoten im Rahmen um } U) \\
 &4j \cdot (m-1)+ && \text{(Höhe der } Y_{i,j}) \\
 &(3m-j) \cdot 2 \cdot 2+ && \text{(Ketten in } X_{1,j} \text{ und } X_{m,j}) \\
 &(3m-j) \cdot 4 \cdot (m-2)+ && \text{(Ketten in } X_{2,j} \text{ bis } X_{m-1,j}) \\
 &4 \cdot (m-1)+ && \text{(quer gezeichnete } X_{i,j}) \\
 &5 \cdot (1) && \text{(hoch gezeichnetes } X_{i,j}) \\
 &= 2m + 4jm - 4j + 12m - 4j + \\
 &12m^2 - 24m - 4jm + 8j + 4m - 4 + 5 \\
 &= 12m^2 - 6m + 1
 \end{aligned}$$

Damit ist $L' \leq H$.

Der Graph V enthält neben den vertikalen Pfaden, die Bedingung (3) erfüllen, auch noch andere Pfade. Wir zeigen auch für solche Pfade, dass ihre Höhe kleiner gleich H ist.

Sei E' die Menge der senkrechten Kanten um die ungefüllten Felder in V , die zum Ausgleich der verschiedenen Höhen der $X_{i,j}$ geschaffen wurden. Auf Grund der Konstruktion von V muss ein Pfad, der (3) nicht erfüllt, durch eine Kante aus E' verlaufen.

In Abbildung 8.9 ist das durch die beiden durchgezogenen Pfeile dargestellt. Es genügt damit zu zeigen: Die Längen der Pfade bis zu den ungefüllten Boxen, die die obige Bedingung (4) erfüllen, sorgen dafür, dass bei den ungefüllten Boxen die senkrechten Kanten mindestens Länge eins haben. Damit legen die Pfade, die Bedingung (4) erfüllen, die Höhe einer Box im Gesamtgraphen fest.

Wir zeigen also: Sei $1 \leq i \leq m-1$ und $1 \leq j \leq 3m-1$.

(5) Es liegen alle Knoten am unteren Rand der Box $X_{i,j}$ um mindestens eins niedriger als die Knoten am unteren Rand der Box $X_{i,j+1}$.

(6) Es liegen alle Knoten am unteren Rand der Box $Y_{i,j}$ um mindestens eins höher als die Knoten am unteren Rand der Box $Y_{i,j+1}$.

zu (5): Nach Behauptung 1 gilt für die Länge l_1 des Pfades vom oberen Ende des Graphen zu den unteren Knoten der Box $X_{i,j}$ und für die Länge l_2 des Pfades vom oberen Ende des Graphen zu den unteren Knoten der Box $X_{i,j+1}$:

$$l_1 \geq 12mi + 6i - 6m - 2j \text{ und } l_2 \leq 12mi + 6i - 6m - 2(j+1) + 1 = 12mi + 6i - 6m - 2j - 1.$$

Damit gilt (5).

zu (6): Nach Behauptung 2 gilt für die Länge l_1 des Pfades vom oberen Ende des Graphen zu den unteren Knoten der Box $Y_{i,j}$ und für die Länge l_2 des Pfades vom oberen Ende des Graphen zu den unteren Knoten der Box $Y_{i,j+1}$:

$$l_1 \leq 12mi + 6i - 6m + 2j \text{ und } l_2 \geq 12mi + 6i - 6m + 2(j+1) - 1 = 12mi + 6i - 6m + 2j + 1$$

Damit gilt (6). ■

Das Problem, für eine gegebene orthogonale Repräsentation im 4-Sektorenmodell eine flächenminimale Zeichnung zu generieren, ist also NP-vollständig. Damit gibt es keine effiziente Lösung und es muss nach Heuristiken zum Zeichnen gesucht werden. Da wir als Zwischenschritt eine orthogonale Repräsentation konstruieren, sind hier zunächst alle

Standardalgorithmen (siehe z. B. [FK95, FHK98]) anwendbar. Trotzdem haben wir uns hier entschieden, eine eigene Zeichenroutine zu entwickeln. Da hier eine eingeschränkte Klasse von orthogonalen Repräsentationen berücksichtigt werden muss, kann der Algorithmus einfach und damit unter bestimmten Gesichtspunkten variabler gestaltet werden. Wir stellen unsere Zeichenroutine im nächsten Abschnitt vor.

8.2 Algorithmus zum Berechnen einer Zeichnung

Für die Zeichenroutine soll als Eingabe eine orthogonale Repräsentation des Graphen vorliegen. Der Algorithmus aus Kapitel 6 liefert eine solche. Falls mit dem Algorithmus zur Knickoptimierung aus Kapitel 7 Kanten berechnet werden, die abweichend vom ursprünglichen Kantenverlauf senkrecht, waagrecht oder mit einem anstatt mit zwei Knicken gezeichnet werden sollen, dann muss die orthogonale Repräsentation entsprechend angepasst werden. Dieser Schritt ist einfach und wir lassen ihn hier weg.

Die Ordnungsrelationen \preceq_x und \preceq_y des Restriktionensystems definieren je eine Halbordnung der Knoten in x- und y-Richtung.⁹ Eine Halbordnung kann auch als DAG interpretiert werden und damit besteht der Zeichenalgorithmus eigentlich nur aus einer Anwendung des Schrittes zur Zeilenberechnung aus dem Sugiyama-Algorithmus ([STM81]) in x- und y-Richtung zur Zuweisung der Zeilen und Spalten. Damit eine planare Zeichnung garantiert werden kann, werden vorher noch zusätzliche Lagebeziehungen eingefügt. Zuletzt werden die Kantenverläufe festgelegt.

Der Vorteil von diesem Zeichenalgorithmus ist, dass er leicht um minimale Abstände zwischen Knoten erweitert werden kann. Wir werden später noch sehen, dass sich variable Knotengrößen und Kantenbeschriftungen einfach in den Ansatz integrieren lassen.

Die Eingabe für den Algorithmus sei also ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem und eine daraus berechnete orthogonale Repräsentation. Der Algorithmus teilt sich in insgesamt fünf Schritte, die im Folgenden erläutert werden:

1. Ergänzen der Kanten mit zwei Knicken um Dummyknoten.
2. Sicherstellen der Planarität durch zusätzliche Lageabhängigkeiten.
3. Festlegen der Zeilen und Spalten der Knoten.
4. Berechnen der Knotenkoordinaten.
5. Festlegen der Kantenverläufe.

Wir stellen die Schritte im Folgenden vor.

1. Ergänzen der Kanten mit zwei Knicken um Dummyknoten.

Wenn eine Kante zwei Knicke hat, also Element von $E_2 \setminus E^1$ ist, dann wird für den senkrechten Kantenabschnitt eine eigene Spalte benötigt. Das heißt, die Endknoten der

⁹ Wenn das keine Halbordnungen wären, dann gäbe es keine Zeichnung, die das Restriktionensystem respektiert.

Kante dürfen nicht in aufeinander folgenden Spalten liegen, sondern es muss noch mindestens eine Spalte zwischen ihnen existieren.

Unser Algorithmus platziert, wie der Algorithmus von Sugiyama, einen Knoten in einer Spalte¹⁰, wenn alle seine Vorgänger platziert sind. Deshalb ersetzen wir für die Berechnung der Spalten der Knoten eine Kante $e = \{v_1, v_2\}$ mit zwei Knicken durch zwei Kanten $e_1 = \{v_1, v\}$ und $e_2 = \{v, v_2\}$ und den neuen Knoten v zwischen den Kanten. Der neue Knoten v sorgt dann dafür, dass zwischen v_1 und v_2 noch mindestens eine Spalte liegen muss.

Der senkrechte Abschnitt der Kante e wird dann in der Spalte, in der der Knoten v liegt, platziert. Nach der Berechnung der Spalten werden diese Knoten wieder gelöscht und der Originalgraph wird wieder hergestellt. Nur die für den Knoten berechnete Spalte wird für den senkrechten Kantenabschnitt verwendet.

2. Sicherstellen der Planarität durch zusätzliche Lageabhängigkeiten.

Wir müssen beim Zeichnen des Graphen sicherstellen, dass die Zeichnung planar ist und die Einbettung nicht verändert wird. Die beiden Halbordnungen in der bisherigen Form können das nicht garantieren, wie am Beispiel in Abbildung 8.10 zu sehen ist. Abbildung 8.10(a) zeigt das Restriktionensystem eines Graphen, Abbildung 8.10(b) eine Zeichnung, die bzgl. dieses Restriktionensystems legal ist. Die Zeichnung ist aber nicht planar.

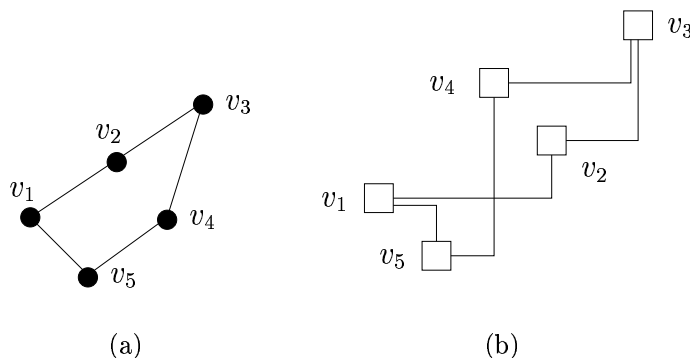


Abbildung 8.10: Beispiel, dass das Restriktionensystem und die orthogonale Repräsentation keine planare Zeichnung garantieren.

Deshalb ergänzen wir die Lageabhängigkeiten in x- und y-Richtung in jeder Fläche um zusätzliche Abhängigkeiten. Sie trennen die vier Sektoren einer Fläche, so dass sich Kanten in verschiedenen Sektoren nicht schneiden können, außer sie liegen an einem gemeinsamen Endknoten. Diese Kanten stellen sicher, dass nur noch durch die „Bend-or-end“ Eigenschaft des Kandinsky-Modells ein Kantenschnitt entstehen kann. Auch dort werden wir mit zusätzlichen Lageabhängigkeiten die Planarität sicherstellen. Da wir

¹⁰ Bei Sugiyama werden die Knoten in der Regel zeilenweise platziert.

diesen Schritt für alle Flächen des Graphen durchführen, ist die generierte Zeichnung des Graphen automatisch planar. Kanten, die im gleichen Sektor liegen, können sich nicht schneiden, da sie immer auf verschiedenen Seiten eines Knotens liegen müssen.

In Abbildung 8.11 ist ein Beispiel dargestellt. Die beiden Pfeile symbolisieren die neuen Lageabhängigkeiten in y -Richtung. Die neuen Lageabhängigkeiten stellen sicher, dass alle Knoten in Sektor 1 oberhalb der Knoten in Sektor 3 liegen müssen. Außerdem müssen alle Knoten in Sektor 2 oberhalb aller Knoten in Sektor 4 liegen.

Wenn einzelne Sektoren leer sind, dann müssen kleinere Änderungen zur Abbildung vorgenommen werden. Wenn z.B. Sektor 2 leer ist, dann muss Sektorstartknoten 3 oberhalb der Knoten in Sektor 1 liegen. Im Lemma unten werden diese Fälle mit berücksichtigt.

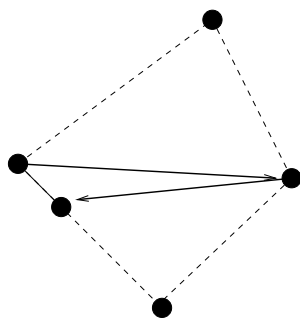


Abbildung 8.11: Trennung von verschiedenen Sektoren einer Fläche, um die Planarität der Zeichnungen zu garantieren.

Was bewirken diese Lageabhängigkeiten, wenn man ignoriert, dass bedingt durch die „bend-or-end“-Eigenschaft des Kandinsky-Modells an einem Knoten eine Kante ohne Knick einen Kantenschnitt verursachen kann?

Es gilt auch ohne die neuen Lagebeziehungen, dass sich Kanten in Sektor 1 und 4 und in Sektor 2 und 3 nicht schneiden können, denn sie sind immer in y -Richtung durch mindestens einen dazwischen liegenden Knoten getrennt. Ebenso können sich Kanten in Sektor 1 und 2 und in Sektor 3 und 4 nicht schneiden. Damit können sich nur noch Kanten in Sektor 1 mit Kanten in Sektor 3 und Kanten in Sektor 2 mit Kanten in Sektor 4 schneiden. Die neuen Lageabhängigkeiten trennen diese Sektoren aber in y -Richtung, so dass es auch hier zu keinen Schnitten kommen kann.

Es kann sein, dass Flächen höher gezeichnet werden, als unbedingt notwendig. Das recht einfache Verfahren ist uns aber diesen Nachteil wert.

Die zusätzlichen Abhängigkeiten werden nach dem Berechnen der Zeilen und Spalten wieder entfernt. Damit haben sie bei späteren Einfügeoperationen keinen Einfluss.

Wir fügen zur Darstellung der Lageabhängigkeiten neue Kanten in den Graphen ein, bei denen jeweils nur entweder das Attribut \preceq_x oder das Attribut \preceq_y belegt wird. Diese Kanten sollen nur dann bei der Berechnung der Zeilen und Spalten berücksichtigt werden, wenn das entsprechende Attribut belegt ist. Also bei der Berechnung der Zeilen, falls \preceq_y belegt ist und bei der Berechnung der Spalten sonst.

Das folgende Lemma zeigt, dass diese Vorgehensweise eine planare Zeichnung garantiert.

Lemma 63

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R . Seien außerdem $E^W \subset E$, $E^s \subset E$ und $E^1 \subset E$ gegeben.

Jede Fläche wird wie folgt um Lageabhängigkeiten ergänzt:

Es wird sichergestellt, dass die Knoten in Sektor 1 oberhalb der Knoten von Sektor 3 und die Knoten in Sektor 2 oberhalb der Knoten von Sektor 4 liegen müssen. Wenn ein Sektor i leer ist, dann ist das für den Sektorstartknoten $i + 1$ nicht möglich (wenn z.B. Sektor 2 leer ist, dann muss Sektorstartknoten 3 oberhalb aller Knoten in Sektor 1 liegen). In diesem Fall werden die obigen Eigenschaften nur für die Sektormittelknoten von Sektor $i + 1$ sichergestellt.

Seien für alle Kanten e aus E^W der zyklische Vorgänger und Nachfolger an den beiden Endknoten gegeben. Falls der Vorgänger oder Nachfolger auf der gleichen Seite des Endknotens endet wie e , dann seien die Lageabhängigkeiten in x -Richtung so ergänzt, dass der Kantenknick des Vorgängers oder Nachfolgers näher am Endknoten liegen muss als der zweite Endknoten von e .

Seien für alle Kanten e aus E^s der zyklische Vorgänger und Nachfolger an den beiden Endknoten gegeben. Falls der Vorgänger oder Nachfolger auf der gleichen Seite des Endknotens endet wie e , dann seien die Lageabhängigkeiten in y -Richtung so ergänzt, dass der Kantenknick des Vorgängers oder Nachfolgers näher am Endknoten liegen muss als der zweite Endknoten von e .

Eine Zeichnung des Graphen, die das Restriktionensystem schwach respektiert und in der die Kanten aus E^W waagerecht ohne Knick und die Kanten aus E^s senkrecht ohne Knick gezeichnet werden, ist planar.

BEWEIS Wenn jede Fläche eines Graphen planar gezeichnet ist, dann ist auch der gesamte Graph planar gezeichnet. Es genügt also, die Behauptung für eine beliebige Fläche zu zeigen.

Kanten, die im gleichen Sektor liegen, können sich nicht schneiden. Sie sind immer durch mindestens einen Knoten getrennt.

Kanten, die in verschiedenen Sektoren liegen und nicht auf der gleichen Seite eines Knotens beginnen, können sich ebenfalls nicht schneiden, denn auch sie sind durch mindestens einen Knoten getrennt.

Es bleiben Kanten, die auf der gleichen Seite eines Knotens beginnen. Wenn sie zu verschiedenen Seiten abknicken, dann können sie sich nicht schneiden. Wenn eine Kante ohne Knick gezeichnet wird, dann können sie sich ebenfalls nicht schneiden, denn in diesem Fall stellen zusätzliche Lagebeziehungen sicher, dass der Knick der Kante mit Knick näher am Endknoten liegt als der Endknoten der Kante ohne Knick.

Es bleibt der Fall, dass beide Kanten auf der gleichen Seite eines gemeinsamen Endknotens v enden und zur gleichen Seite abknicken. Das kann nur in 4 Fällen sein:

1. Die in der zyklischen Reihenfolge um v erste der beiden Kanten e_1 ist die erste in Sektor 1 und die zweite Kante e_2 ist die letzte in Sektor 3 und beide Kanten

verlassen v nach rechts oben. Dann ist aber der zweite Endknoten von e_1 links von dem zweiten Endknoten von e_2 und die Kanten schneiden sich nicht.

2. Die in der zyklischen Reihenfolge um v erste der beiden Kanten e_1 ist die erste in Sektor 2 und die zweite Kante e_2 ist die letzte in Sektor 4 und beide Kanten verlassen v nach rechts unten. Dann ist aber der zweite Endknoten von e_2 links von dem zweiten Endknoten von e_1 und die Kanten schneiden sich nicht.
3. Die in der zyklischen Reihenfolge um v erste der beiden Kanten e_1 ist die erste in Sektor 3 und die zweite Kante e_2 ist die letzte in Sektor 1 und beide Kanten verlassen v nach links unten. Dann ist aber der zweite Endknoten von e_2 unterhalb von dem zweiten Endknoten von e_1 und die Kanten schneiden sich nicht.
4. Die in der zyklischen Reihenfolge um v erste der beiden Kanten e_1 ist die erste in Sektor 4 und die zweite Kante e_2 ist die letzte in Sektor 2 und beide Kanten verlassen v nach links oben. Dann ist aber der zweite Endknoten von e_2 oberhalb von dem zweiten Endknoten von e_1 und die Kanten schneiden sich nicht.

Es kann also keinen Kantenschnitt geben. ■

Zur Berechnung der zusätzlichen Lagebeziehungen müssen immer nur lokale Eigenschaften an einem Knoten oder in einer Fläche für wenige ausgewählte Knoten überprüft werden. Deshalb können diese Lageabhängigkeiten in linearer Zeit berechnet werden.

3. Festlegen der Zeilen und Spalten.

Bei der Festlegung der Zeilen und Spalten, in denen die Knoten platziert werden, wollen wir jeweils alle Knoten in die Zeile oder Spalte legen, deren Vorgänger bzgl. der Halbordnung in x - bzw. y -Richtung schon platziert wurden. Jeder Knoten wird also so früh wie möglich eingeordnet.

Dabei ergeben sich allerdings Probleme, wenn Kanten senkrecht oder waagrecht ohne Knick gezeichnet werden sollen. Denn es kann sein, dass die Endknoten der Kante auf diese Weise nicht in der gleichen Zeile oder Spalte gelegt werden. Deshalb werden alle Knoten, die in der gleichen Zeile oder Spalte liegen sollen, vor der Berechnung der Zeilen oder Spalten temporär durch einen gemeinsamen Repräsentanten ersetzt. Dieser erbt alle Vorgänger in x - bzw. y -Richtung von allen seinen repräsentierten Knoten.

Die Repräsentanten können in linearer Zeit berechnet werden. Dazu wird zunächst der Graph kopiert und dann alle Kanten gelöscht, die keiner waagerechten (bzw. zur Berechnung der Spalten senkrechten) Kante entsprechen. Dann müssen alle Zusammenhangskomponenten des übrigen Graphen in einer Zeile (bzw. Spalte) platziert werden.

Bei der Berechnung der Zeilen und Spalten ignorieren wir die Knoten, für die es einen Repräsentanten gibt. Statt dessen berechnen wir die Zeile oder Spalte des Repräsentanten.

Zur Berechnung der Zeilen bzw. Spalten wird die Menge der Lagebeziehungen wieder als Halbordnung interpretiert. Dafür kann eine topologische Sortierung berechnet werden. Diese entspricht dann der Zuordnung der Knoten zu Zeilen und Spalten. Eine

topologische Sortierung kann in Linearzeit berechnet werden. Dies genügt, da die Knoten uniforme Größe haben.

Jeder Knoten erhält als Ergebnis dieses Schrittes eine Zeile und Spalte zugewiesen, die in den Attributen $v.col$ und $v.row$ gespeichert werden.

4. Berechnen der Knotenkoordinaten.

Wir berechnen die Knotenpositionen direkt aus den Zeilen und Spalten eines Knotens, also $v.x = v.col \cdot \delta$ und $v.y = v.row \cdot \delta$, wobei δ wieder der Abstand zwischen zwei Gitterlinien ist. Die Komplexität dieses Schrittes ist $O(n)$.

5. Festlegen der Kantenverläufe.

Bisher wurden die Positionen der Knoten berechnet. Zusätzlich kennen wir die zyklische Ordnung der Kanten um die Knoten. Außerdem wissen wir, auf welcher Seite eines Knotens eine Kante endet.

Im Kandinsky-Modell endet jede Kante ohne Knick in der Mitte eines Knotens. Alle anderen Kanten verteilen sich gleichmäßig auf den beiden Seiten der Senkrechte. Aus der zyklischen Ordnung der Kanten um die Knoten kann leicht die Reihenfolge, in der die Kanten auf einer Seite eines Knotens enden müssen, berechnet werden. Die Positionen von Kantenknicken ergeben sich aus den Endpunkten an den Knoten. Bei einer Kante mit zwei Knicken wird das senkrechte Kantenstück in der Spalte platziert, die dem Knoten zugewiesen wurde, der zur Repräsentation dieses Kantensegments in den Graphen eingefügt wurde.

Zur Berechnung der Kantenverläufe müssen die Kanten um jeden Knoten einmal durchlaufen werden. Damit erhalten wir in Summe lineare Laufzeit.

Damit haben wir eine einfache Heuristik zum Zeichnen des Graphen aus einer orthogonalen Repräsentation, die dem 4-Sektorenmodell entspricht, beschrieben. Jeder der Schritte benötigt lineare Laufzeit. Da jeder Schritt in $O(n)$ durchgeführt werden kann, erhalten wir als Gesamtkomplexität der Zeichenroutine eine lineare Laufzeit.

9 Erweiterungen des Basisalgorithmus

In den bisherigen Kapiteln haben wir ein Verfahren für planare, orthogonale Zeichnungen von zweifach zusammenhängenden Graphen erarbeitet. In diesem Abschnitt wollen wir uns mit der Integration von zusätzlichen Erweiterungen in das bisherige Verfahren beschäftigen. Dabei streben wir keine im Vergleich zur Literatur herausragenden Ergebnisse an. Das Ziel ist es zu zeigen, dass das beschriebene Verfahren um zusätzliche Anforderungen aus der Praxis relativ einfach erweitert werden kann.

Wir betrachten zunächst beliebige Knotengrößen und Kantenbeschriftungen. Ähnlich wie benutzerdefinierte Knotengrößen, die mit Hilfe von Knotenbreite und -höhe gelöst werden, arbeiten wir bei Kantenbeschriftungen mit einer minimalen Länge einer Kante in x- und y-Richtung. Diese Information wird in den Attributen $e.w$ und $e.h$ gespeichert.

Anschließend beschäftigen wir uns mit einfach zusammenhängenden Graphen. Wir führen dabei einen einfach zusammenhängenden Graphen wieder in einen zweifach zusammenhängenden über, so dass alle anderen Algorithmen unverändert übernommen werden können. Als Letztes betrachten wir noch nicht planare Graphen.

9.1 Benutzerdefinierte Knotengrößen

Im letzten Kapitel wurden für den Fall von uniformen Knotengrößen die Zeilen und Spalten berechnet. Wir wollen allerdings auch für benutzerdefinierte Knotengrößen die Zeilen und Spalten eines Knotens berechnen können. In diesem Fall ist es möglich, dass sich ein Knoten über mehrere Zeilen oder Spalten erstreckt.

Es müssen hier also zusätzlich die Breite und Höhe eines Knotens berücksichtigt werden. Sei $v.x$ die x-Koordinate und $v.y$ die y-Koordinate des Knotenmittelpunktes. Außerdem wird durch das Attribut $v.h$ die Höhe und durch $v.w$ die Breite des Knotens v definiert. Damit erhalten wir $v.x - v.w/2$ als x-Koordinate der linken Seite des Knotens v . Alle anderen Seiten ergeben sich entsprechend.

Sei δ der Abstand zwischen je zwei Gitterlinien und min_e die Minimallänge eines Kantensegmentes. Ein Knoten v darf nur dann in einer Spalte („col“) platziert werden wenn

1. alle Vorgänger platziert sind und
2. für alle Vorgänger v' von v gilt:
$$col \cdot \delta \geq v'.col() \cdot \delta + v'.w/2 + v.w/2 + min_e$$

Analoges gilt für die Berechnung der Zeilen.

Bei der zweiten Bedingung wird einfach ausgerechnet, ob zwischen linkem und rechtem Rand von je zwei adjazenten Knoten ein Minimalabstand eingehalten wird.

Die Laufzeit für die Berechnung ergibt sich aus dem Maximum aus

1. der Summe der Eingangsgrade der Knoten und
2. der Anzahl der besuchten Zeilen (bzw. Spalten).

Punkt 1 ergibt sich daraus, dass ein Knoten platziert werden kann, wenn alle seine Vorgänger platziert sind. Das kann in einer einfachen Zählvariable verwaltet werden.

Der zweite Punkt ist hier deshalb wichtig, weil nie sichergestellt wird, dass auf jeder Zeile oder Spalte ein Knotenmittelpunkt liegt. Wir gehen aber bei realen Graphen davon aus, dass die Größe der Knoten nicht mit der Anzahl an Knoten wächst, sondern eine konstante Obergröße $\delta \cdot k$ hat. Dann wird spätestens in jeder k -te Zeile (bzw. Spalte) ein Knoten platziert und die Laufzeit ist wieder linear. Die Laufzeit entspricht damit auch dem Zuweisen von Zeilen an die Knoten bei einem Sugiyama-Zeichenverfahren, wenn dabei verschiedene Knotengrößen betrachtet werden (vgl. z.B. [Sch01]).

9.2 Kantenbeschriftungen

Wie wir bei der Berechnung der Knotenkoordinaten gesehen haben, ist unser Algorithmus auch dann anwendbar, wenn die Höhe und Breite der Knoten ein Parameter der Eingabe sind. Die zusätzliche Anforderung, verschieden große Knoten verarbeiten zu können, ergibt sich oft aus der Beschriftung der Knoten. Manchmal ist es zusätzlich noch nötig, auch Kanten mit Texten zu versehen. Deshalb sollen Kantenbeschriftungen in das Verfahren integriert werden. Eine Beschriftung der Kanten ist nur dann sinnvoll, wenn sie folgende Eigenschaften erfüllt (vgl. z.B. auch [Yoe72, Imh75]):

1. Die Kantenbeschriftungen sind gut leserlich. Deshalb müssen sie überschneidungsfrei gezeichnet werden und dürfen auch von keiner Kante geschnitten werden oder von einem Knoten überlagert werden.
2. Die Kantenbeschriftungen müssen einer Kante eindeutig zugeordnet werden können und diese Zuordnung sollte intuitiv verständlich sein.
3. Ebenso wie Knotenbeschriftungen sollen auch Kantenbeschriftungen in Höhe und Breite variieren dürfen.

Algorithmen zur Beschriftung von Kanten gehen üblicherweise so vor, dass zunächst eine Zeichnung des Graphen berechnet wird, danach werden die Kantenbeschriftungen platziert. An einer Kante gibt es mehrere Positionen, an denen die Beschriftungen angebracht werden dürfen. Diese Positionen werden als verschieden gut bewertet und erhalten

damit ein unterschiedliches Gewicht. Gesucht ist dann eine Zuordnung der Kantenbeschriftungen mit optimalem Gewicht. In [KT96] wird bewiesen, dass dieses Problem NP-vollständig ist.

Unser Ansatz unterscheidet sich davon. Wir definieren, an welchen Stellen Kanten beschriftet werden dürfen. Dann greifen wir so weit in die übrigen Algorithmen ein, dass eine planare Zeichnung mit den gewählten Positionen für die Beschriftungen möglich ist. Wir werden sehen, dass das die Knickoptimierung und die Berechnung der Zeilen und Spalten der Knoten ist.

Die einzige uns bekannte Arbeit, die das Erzeugen des Graphen aus der orthogonalen Repräsentation und das Beschriften der Kanten gemeinsam betrachtet ist [KM99a]. Die Autoren präsentieren einen „branch-and-cut“ Algorithmus, der eine Zeichnung des Graphen mit optimal beschrifteten Kanten erzeugt.

Unser Algorithmus erzeugt zwar im Gegensatz zu [KM99a] keine optimalen Zeichnungen, dafür ändert sich die Laufzeit unserer Algorithmen nicht und wir erhalten Algorithmen mit, im Falle einer kombinierten Knickoptimierung, einer Laufzeit von $O(n^2 \log n)$.

Definition 26 ($e.lw$, $e.lh$)

Gegeben sei ein Graph $G = (V, E)$. Dann beschreibt das Attribut $e.lw$ die Breite und $e.lh$ die Höhe der Beschriftung der Kante e . ■

Die Attribute $e.lw$ und $e.lh$ beschreiben die Größe der Label, die zu einer Kante gehören. Die minimale Länge und Höhe einer Kante weichen in der Regel davon ab, wie wir noch sehen werden.

Definition 27 ($e.w$, $e.h$)

Gegeben sei ein Graph $G = (V, E)$. Dann beschreibt das Attribut $e.w$ die minimale Länge des waagerechten Kantenstückes und $e.h$ die minimale Länge des senkrechten Kantenstückes von e . ■

Man beachte in der letzten Definition, dass wir nicht unterscheiden, ob eine Kante einen, keinen oder zwei Knicke hat. Das ist deshalb nicht nötig, da wir einer Kante ohne Knick nur eines von den beiden Attributen zuweisen. Und bei einer Kante mit zwei Knicken wird diese zur Berechnung der Spalten der Knoten durch zwei Kanten ersetzt, so dass wir wieder nur ein waagerechtes Kantenstück erhalten.

Ziel dieses Abschnittes ist es, die Kantenattribute $e.w$ und $e.h$ so zu bestimmen, dass die oben geforderten Eigenschaften erfüllt werden. Dazu untersuchen wir nacheinander die verschiedenen möglichen Kantenverläufe und zeigen jeweils, wie die Kantenbeschriftung erfolgt. Im gesamten Kapitel ignorieren wir die Knotengrößen. Sie müssen einfach nur zu den Minimalgrößen für die Kantenbeschriftungen gezählt werden, wenn ein Teil der Endknoten der zu beschriftenden Kante in dem Rechteck liegen kann, das für die Kantenbeschriftung verwendet werden soll.

Außerdem kann es passieren, dass bei einer Kante mehrfach eine Zuweisung an eines der Attribute $e.w$ und $e.h$ erfolgt. Dann werde dem Attribut immer das Maximum aus dem bisherigen und dem neuen Wert zugewiesen, ohne dass das nochmals erwähnt wird.

9.2.1 Kanten mit einem Knick

Platzierung des Kantenlabel

O.B.d.A¹ gehen wir stets davon aus, dass die zu beschriftende Kante e auf der rechten Seite des linken Knotens beginnt. Danach folgt ein senkrecht Kantenstück, das auf der Oberseite des rechten Endknotens endet. Stets sei v_1 der linke Endknoten der zu beschriftenden Kante und v_2 der rechte.

Die Kante wird immer auf der Innenseite des Knickes beschriftet. Unserer Meinung nach ist die Zuordnung der Beschriftung zur Kante damit am einfachsten. In Abbildung 9.1 ist die Beschriftung einer Kante dargestellt.

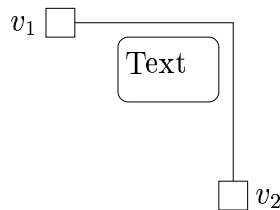


Abbildung 9.1: Platzierung der Kantenbeschriftung bei Kanten mit einem Knick.

Raum für Beschriftung sicherstellen

Wir ignorieren zunächst den Fall, dass auf der Unterseite von v_1 eine senkrechte und auf der linken Seite von v_2 eine waagerechte Kante ohne Knick enden kann. Wir werden bei diesen Kantenverläufen dann noch sehen, dass es hier zu Überlagerungen von Beschriftungen kommen kann und wie dieses Problem gelöst werden kann.

Wie kann der Raum für die Kantenbeschriftung sichergestellt werden? Die Kante e liegt in der Fläche f unterhalb der Kante in Sektor 2. Wir haben bei der Berechnung der Zeilen und Spalten der Knoten festgestellt, dass alle Knoten und Kanten, die in Sektor 1 liegen, links von v_1 liegen müssen. Genauso müssen alle Knoten und Kanten in Sektor 3 unterhalb von v_2 liegen.

Die Beschriftung wird in dem Rechteck, das durch die Knoten v_1 und v_2 definiert wird, eingefügt. Damit kann ein Knoten oder eine Kante aus der Fläche f nur dann durch die Beschriftung gehen, wenn er in Sektor 4 liegt. Wir haben aber bei der Berechnung der Zeilen und Spalten durch zusätzliche Lageabhängigkeiten sichergestellt, dass alle Sektormittelknoten in Sektor 4 unterhalb der Knoten in Sektor 2 liegen, und damit auch unterhalb v_2 .

Also können nur dann Knoten und Kanten von Sektor 4 in dem durch die beiden Knoten beschriebenen Rechteck liegen, wenn einer der beiden Knoten v_1 oder v_2 ein Sektorstartknoten ist. Wenn v_1 Sektorstartknoten ist, dann muss dabei der zyklische Nachfolger von e beim Knoten v_1 in f die letzte Kante in Sektor 4 sein. Sei v_3 der zweite

¹ Die Allgemeinheit folgt daraus, dass alle Überlegungen spiegelsymmetrisch bzgl. x- und y-Achse sind.

Endknoten dieser Kante. Wenn v_2 der Sektorstartknoten ist, dann muss der zyklische Vorgänger von e am Knoten v_2 in f die erste Kante in Sektor 4 sein. Sei v_4 der zweite Endknoten dieser Kante. Wenn keine dieser beiden Bedingungen zutrifft, dann gilt wieder das obige Argument und keine Knoten und Kanten in Fläche f können durch die Beschriftung schneiden. Wir werden die Kantenbeschriftung in diesen Fällen in dem Rechteck platzieren, das durch den Kantenknick von e und Knoten v_3 bzw. Knoten v_4 bestimmt wird. Auf Grund des 4-Sektorenmodells kann in diesem Rechteck kein weiterer Knoten liegen.

Die Alternativen zur Platzierung des Labels sind in Abbildung 9.2 dargestellt.

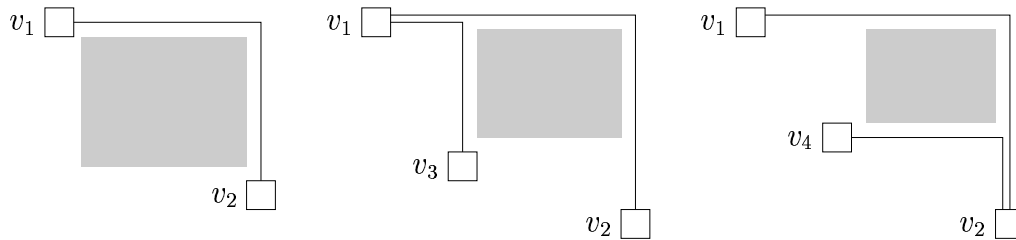


Abbildung 9.2: Alternativen für die Sicherstellung des Raumes für die Kantenbeschriftung bei Kanten mit einem Knick.

Die notwendige Breite und Höhe erhalten wir wie folgt:

1. Wenn v_1 und v_2 keine Sektorstartknoten sind, dann setzen wir zur Berechnung der Zeilen und Spalten $e.w = e.lw$ und $e.h = e.lh$.
2. Andernfalls ist einer der beiden Knoten Sektorstartknoten.

Sei v_1 der Sektorstartknoten. Dann sei $e' = \{v_1, v_3\}$ der zyklische Nachfolger von e am Knoten v_1 . Wir setzen zur Berechnung der Zeilen und Spalten $e'.h = e.lh$ und erzeugen eine neue Kante $e'' = \{v_3, v_2\}$ mit $e''.w = e.lw$.

Andernfalls ist v_2 der Sektorstartknoten. Dann sei $e' = \{v_2, v_4\}$ der zyklische Vorgänger von e am Knoten v_2 . Wir setzen zur Berechnung der Zeilen und Spalten $e'.w = e.lw$ und erzeugen eine neue Kante $e'' = \{v_4, v_1\}$ mit $e''.h = e.lh$.

Wenn beide Knoten Sektorstartknoten sind, dann gilt nach Alternative 1 in Fall 2 $\{v_1, v_3\}.h = e.lh$ und $\{v_2, v_4\}.w = e.lw$, was wieder ausreichend ist.

9.2.2 Kanten mit zwei Knicken

Platzierung des Kantenlabel

O.B.d.A gehen wir davon aus, dass die zu beschriftende Kante auf der rechten Seite des linken Knotens beginnt. Danach folgt ein senkrechttes Kantenstück, das nach unten verläuft. Nach einem weiteren waagerechten Kantenstück endet die Kante an der linken

Seite des rechten Endknotens. Stets sei v_1 der linke Endknoten der zu beschriftenden Kante und v_2 der rechte.

Die Kante wird immer auf der Innenseite des linken Knickes beschriftet, was in Abbildung 9.3 dargestellt ist.

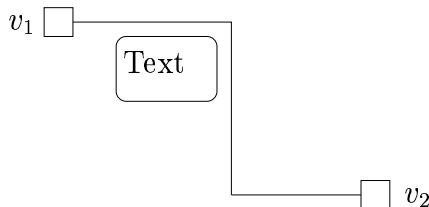


Abbildung 9.3: Platzierung des Kantenlabel bei Kanten mit zwei Knicken.

Raum für Beschriftung sicherstellen

Wieder wird zunächst der Fall ignoriert, dass auf der Unterseite von v_1 eine senkrechte und auf der linken Seite von v_2 eine waagerechte Kante enden kann.

Mit der gleichen Argumentation wie bei Kanten mit einem Knick gilt: Es können nur dann Knoten und Kanten aus f in dem durch die beiden Endknoten der Kante beschriebenen Rechteck liegen, wenn einer der beiden Knoten v_1 oder v_2 ein Sektorstartknoten ist. Und wenn v_1 Sektorstartknoten ist, dann muss dabei der zyklische Nachfolger von e beim Knoten v_1 in f die letzte Kante in Sektor 4 sein. Sei v_3 der zweite Endknoten dieser Kante. Wenn v_2 der Sektorstartknoten ist, dann muss der zyklische Vorgänger von e am Knoten v_2 in f die erste Kante in Sektor 4 sein. Sei v_4 der zweite Endknoten dieser Kante. Wenn keine dieser beiden Bedingungen zutrifft, dann gilt wieder das obige Argument und keine Knoten und Kanten in Fläche f können durch die Beschriftung schneiden.

Wir werden die Kantenbeschriftung in diesen Fällen in dem Rechteck platzieren, das durch den linken Kantenknick von e und Knoten v_3 bzw. Knoten v_4 bestimmt wird. Auf Grund des 4-Sektorenmodells kann in diesem Rechteck kein weiterer Knoten liegen.

Die Alternativen zur Platzierung des Labels sind in Abbildung 9.4 dargestellt.

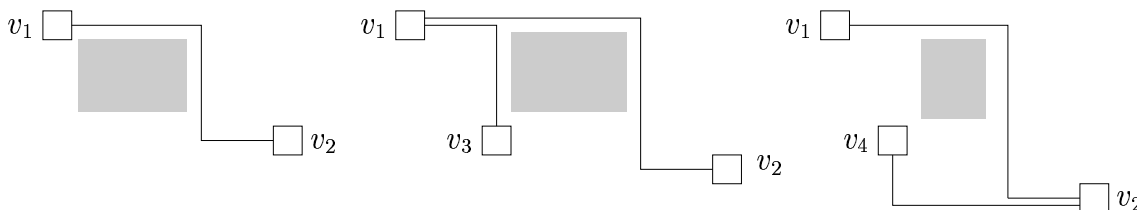


Abbildung 9.4: Alternativen für die Sicherstellung des Raumes für die Kantenbeschriftung bei Kanten mit einem Knick.

Zur Berechnung der Spalten wird bei Kanten mit zwei Knicken für das senkrechte Kantenstück ein Dummyknoten eingefügt. Sei im Folgenden v der Dummyknoten der Kante e . Die notwendige Breite und Höhe erhalten wir wie folgt:

1. Wenn v_1 und v_2 keine Sektorstartknoten sind, dann setzen wir zur Berechnung der Zeilen und Spalten $\{v_1, v\}.w = e.lw$ und $e.h = e.lh$.
2. Andernfalls ist einer der beiden Knoten Sektorstartknoten.

Sei v_1 ein Sektorstartknoten. Dann sei $e' = \{v_1, v_3\}$ der zyklische Nachfolger von e am Knoten v_1 . Wir setzen zur Berechnung der Zeilen und Spalten $e'.h = e.h = e.lh$ und erzeugen eine neue Kante $e'' = \{v_3, v\}$ mit $e''.w = e.lw$.

Andernfalls ist v_2 der Sektorstartknoten. Dann sei $e' = \{v_2, v_4\}$ der zyklische Vorgänger von e am Knoten v_2 . Wir erzeugen eine neue Kante $e_1 = \{v_4, v\}$ und setzen zur Berechnung der Zeilen und Spalten $e_1.w = e.lw$. Außerdem erzeugen wir eine neue Kante $e_2 = \{v_4, v_1\}$ mit $e_2.h = e.lh$.

9.2.3 Kanten ohne Knick

Beschriftung der Kante sicherstellen

Zunächst gilt, dass es sowohl bei waage- als auch bei senkrechten Kanten ohne Knick sein kann, dass kein Platz für die Kantenbeschriftung vorhanden ist. Eine waagerechte Kante kann im Falle einer orthogonalen Repräsentation wie in Abbildung 9.5(a), eine senkrechte kann im Falle einer orthogonalen Repräsentation wie in Abbildung 9.6(a) nicht beschriftet werden.

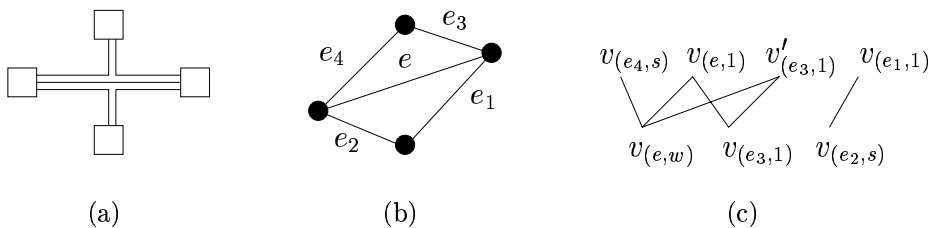


Abbildung 9.5: Situation, in der eine waagerechte Kante nicht beschriftet werden kann.

Beide orthogonalen Repräsentationen können das Ergebnis einer Knickoptimierung mit unseren Algorithmen sein. Ein entsprechendes Restriktionensystem ist den Abbildungen dargestellt, gemeinsam mit dem Graphen G_{opt} , der für die Modellierung des Restriktionensystems verwendet wird.

In so einem Fall kann man den Graphen G_{opt} so anpassen, dass keine orthogonale Repräsentation entstehen kann, die nicht beschriftet werden kann. Dazu kann z.B. in

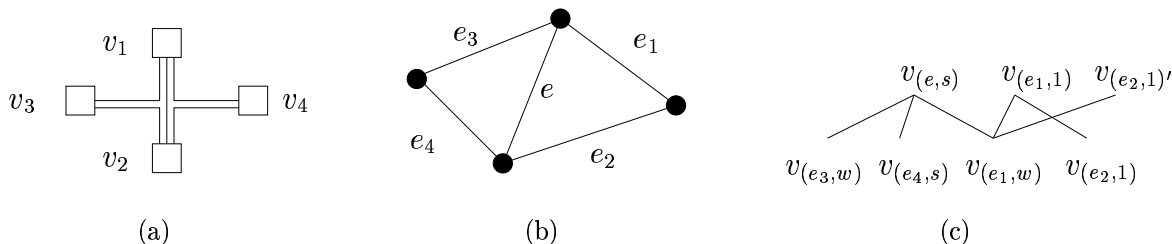


Abbildung 9.6: Situation, in der eine senkrechte Kante nicht beschriftet werden kann.

Abbildung 9.5(c) und 9.6(c) jeweils der Knoten $v_{(e_1,1)}$ gelöscht werden. Man erhält dadurch vielleicht einen Knick mehr, dafür ist sicher gestellt, dass direkt an jeder Kante genügend Platz für die Beschriftung vorhanden ist.

Platzierung der Kantenbeschriftung

In Abbildung 9.5(a) sieht man, dass jeder Teil einer waagerechten Kante ohne Knick durch eine andere Kante für die Beschriftung blockiert sein kann. Jede der vier Kanten kann aber auch fehlen und man kann vor der Berechnung der orthogonalen Repräsentation nicht sagen, wo wirklich Platz für die Beschriftung ist. Entsprechendes gilt für senkrechte Kanten ohne Knick. In so einem Fall wird einfach eine mögliche Stelle ausgewählt.

O.B.d.A.² nehmen wir an, dass wir eine senkrechte Kante $e = \{v_1, v_2\}$ beschriften wollen. Sei v_1 der obere Endknoten und gelte auf Grund des Restriktionensystems $d(e, v_1) = \searrow$. Auf der rechten Seite der Kante sei Platz für die Beschriftung. Die Kante wird dann auf der rechten Seite beschriftet.

Sei außerdem e' der zyklische Nachfolger von e an seinem rechten Endknoten v_2 . Wir betrachten stets die Fläche f , die rechts der Kante e liegt.

Raum für Beschriftung sicherstellen

Es ist möglich, dass Kantenbeschriftungen direkt nebeneinander liegen und damit muss das auch berücksichtigt werden.

Zunächst stellen wir sicher, dass genug Raum zur Platzierung der Kantenbeschriftung von e in x-Richtung vorhanden ist. Abbildung 9.7 illustriert die folgenden Überlegungen.

Wir betrachten die orthogonale Repräsentation, die aus dem Restriktionensystem erzeugt wurde. Wir laufen beginnend bei v_1 im Uhrzeigersinn um die Fläche und suchen das erste Kantensegment, das von links nach rechts verläuft. Ein solches muss es wegen des 4-Sektorenmodells geben. Sei e' die Kante, zu der das Segment gehört. Wenn die Kante nur einen Knick hat, dann sei v ihr rechter Endknoten. Andernfalls sei v der Knoten, der bei der Berechnung der Spalten das senkrechte Kantenstück von e repräsentiert.

² Alle anderen Fälle können durch Spiegelungen daraus abgeleitet werden.

Wir erzeugen zur Berechnung der Spalten eine neue Kante $\{v_1, v\}$ und setzen dann $\{v_1, v\}.w = e.lw$.

Genauso suchen wir beginnend bei v_2 das nächste waagerechte Kantenstück im Gegenuhrzeigersinn. Auch dessen rechten Endpunkt verbinden wir mit einer Kante e'' mit v_2 und setzen $e''.w = e.lw$.

Wegen des 4-Sektorenmodells müssen dann alle Knoten und Kanten zwischen diesen waagerechten Kantenstück rechts neben der Beschriftung liegen, denn sie können nicht weiter links liegen als die Objekte, die mit den beiden neuen Lageabhängigkeiten nach rechts geschoben wurden.

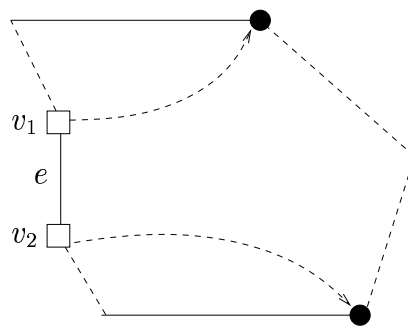


Abbildung 9.7: Sicherstellung des Raumes in x-Richtung bei Kanten ohne Knicke.

Um genügend Raum in y-Richtung sicherzustellen, gehen wir wie folgt vor:

Sei $e' = \{v_1, v_3\}$ der zyklische Vorgänger von e am Knoten v_1 und sei $e'' = \{v_2, v_4\}$ der zyklische Nachfolger von e am Knoten v_2 . Wir betrachten die orthogonale Repräsentation, die aus dem Restriktionensystem berechnet wurde. Dann gibt es fünf Fälle (siehe auch Abbildung 9.8, hier sind die verschiedenen Verläufe beispielhaft für e'' dargestellt), die wir unterscheiden müssen. Sie ergeben sich aus den insgesamt neun Fällen, die sich durch die Kombinationen dieser drei Kantenverläufe für die Kanten e' und e'' erzeugen lassen:

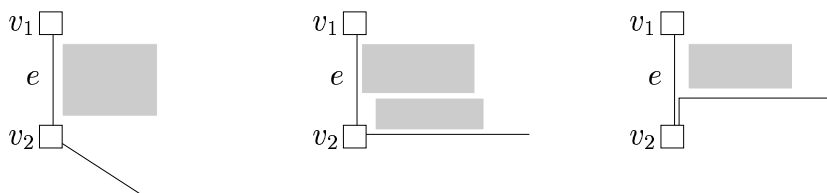


Abbildung 9.8: Sicherstellung des Raumes in y-Richtung bei Kanten ohne Knicke.

1. In jeder Zeichnung, die das Restriktionensystem schwach respektiert, muss v_3 oberhalb von v_1 und v_4 unterhalb von v_2 liegen. In diesem Fall liegt die Beschriftung

von e' oberhalb von v_1 und die Beschriftung von e'' unterhalb von v_2 . Bei der minimalen Höhe von e ist nur die Beschriftung der Kante e zu berücksichtigen, also $e.h = e.lh$.

2. Angenommen, die Kante e' verläuft waagrecht ohne Knick und endet auf der rechten Seite von v_1 . In diesem Fall kann die Beschriftung von e' ebenfalls rechts neben e liegen. Die Beschriftung von e wird dann unterhalb der Beschriftung von e' angeordnet. Bei der minimalen Höhe von e ist die Beschriftung der Kante e' zu berücksichtigen. Wir setzen einen Hilfsknoten $v'_1 = v_1$. Außerdem setzen wir temporär für diesen Zeichenschritt³ $e.lh = e.lh + e'.lh$.
3. Angenommen, die Kante e' verlässt v_1 nach unten, gefolgt von einem waagerechten Kantenstück. Dann muss die Beschriftung von e unterhalb dieses waagerechten Kantenstückes liegen. Wir setzen den Hilfsknoten $v'_1 = v_3$.
4. Angenommen, die Kante e'' verläuft waagrecht ohne Knick und endet auf der rechten Seite von v_2 . In diesem Fall kann die Beschriftung von e'' ebenfalls rechts neben e liegen. Die Beschriftung von e wird dann oberhalb der Beschriftung von e'' angeordnet. Bei der minimalen Höhe von e ist die Beschriftung der Kante e'' zu berücksichtigen. Wir setzen einen Hilfsknoten $v'_2 = v_2$ und temporär für diesen Zeichenschritt $e.lh = e.lh + e''.lh$.
5. Die Kante e'' verlässt v_2 nach oben, gefolgt von einem waagerechten Kantenstück. Dann muss die Beschriftung von e oberhalb dieses waagerechten Kantenstückes liegen. Wir setzen den Hilfsknoten $v'_2 = v_4$.

In den Fällen zwei bis fünf wird noch eine Kante $\{v'_1, v'_2\}$ erzeugt und wir setzen $\{v'_1, v'_2\}.h = e.lh$.

Damit haben wir gezeigt, wie Kantenbeschriftungen in unser Verfahren integriert werden können. Die Integration findet dabei nur durch die Definition von zusätzlichen Minimalabständen statt. Die Ergänzung des Zeichenalgorithmus ist identisch zur Ergänzung für beliebige große Knoten und wird deshalb hier weggelassen.

Im folgenden Abschnitt beschäftigen wir uns mit einer Erweiterung, die in realen Anwendungen wichtig ist. Wir erweitern die betrachtete Graphklasse um einfach zusammenhängende Graphen.

9.3 Einfach zusammenhängende Graphen

In diesem Abschnitt soll untersucht werden, wie einfach zusammenhängende Graphen in unseren Ansatz integriert werden können und welche Probleme dabei auftreten können.

Gegeben sei dabei ein einfach zusammenhängender Graph mit einem Restriktionensystem R . R sei dergestalt, dass jede zweifach Zusammenhangskomponente dem 4-Sektorenmodell entspricht.

³ Nach dem Zeichnen muss der Wert wieder auf den Ausgangswert zurückgesetzt werden.

Wir beschäftigen uns im Folgenden mit zwei Fragestellungen: Gegeben sei ein einfach zusammenhängender Graph mit einem Restriktionensystem. Wie kann der Zeichenalgorithmus so erweitert werden, dass auch dieser Graph gezeichnet werden kann? Und was kann passieren, wenn eine neue Kante zwei zweifach Zusammenhangskomponenten, die bisher nur einfach zusammenhängen, zu einer neuen zweifach Zusammenhangskomponente verbindet?

9.3.1 Zeichnen eines einfach zusammenhängenden Graphen

Zur Vereinfachung fordern wir, dass jede zweifach Zusammenhangskomponente⁴ in einer Fläche einer anderen zweifach Zusammenhangskomponente liegen muss. Wir müssen damit zeigen, wie wir die Komponente in der Fläche mit dem Restgraphen verbinden können, so dass das 4-Sektorenmodell erfüllt ist.

Sei K_1 die äußere und K_2 die innere zweifach Zusammenhangskomponente (siehe auch Abbildung 9.9 für die Bezeichnungen). Sei f die Fläche von K_1 , in die K_2 eingebettet werden soll.

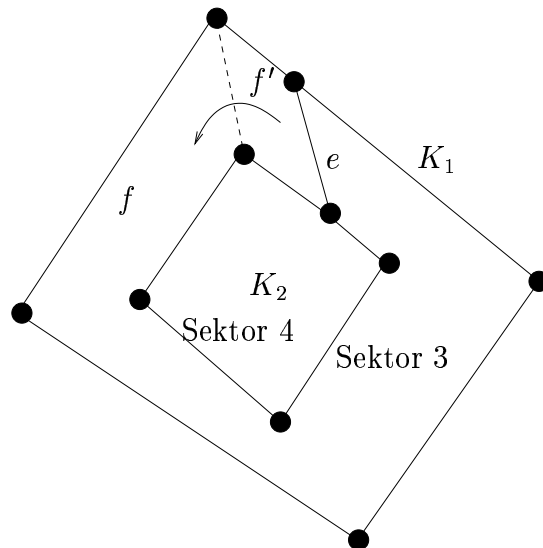


Abbildung 9.9: Illustration des Verbindens von zwei zweifach Zusammenhangskomponenten.

Wir bezeichnen die Sektoren der äußeren Fläche von K_2 genauso wie normalerweise Sektoren einer inneren Fläche bezeichnet werden. Wenn also der zyklische Nachfolger z.B. rechts unterhalb liegt, dann ist ein Knoten Element von Sektor 2. Sektor 3 und 4 sind in Abbildung 9.9 exemplarisch beschriftet.

Sei $e = \{v_1, v_2\}$ die Kante, die die beiden zweifach Zusammenhangskomponenten verbindet.

⁴ Außer einer einzigen, nämlich der äußeren

Dann kann das folgende Verfahren verwendet werden, um die Fläche f in neue Flächen zu partitionieren, die jeweils das 4-Sektorenmodell erfüllen. Dabei werden die Sektorstartknoten der äußeren Komponente geeignet mit den Sektorstartknoten der inneren Komponente verbunden:

Zunächst ignorieren wir die Kante e , die erst im Anschluss integriert wird.

Seien v_1 Sektorstartknoten 1 in Fläche f , v_2 Sektorstartknoten 2 in Fläche f usw., falls der jeweilige Sektorstartknoten existiert (siehe auch Abbildung 9.10). Falls ein Sektorstartknoten i nicht existiert, dann sei v_i ein beliebiger Sektormittelknoten in Sektor $i - 1$. Wenn Sektor $i - 1$ nur den Sektorstartknoten und eine Kante enthält, dann wird v_i nicht belegt (in Abbildung 9.10 auf der rechten Seite wird z.B. v_1 nicht belegt). Dann wird

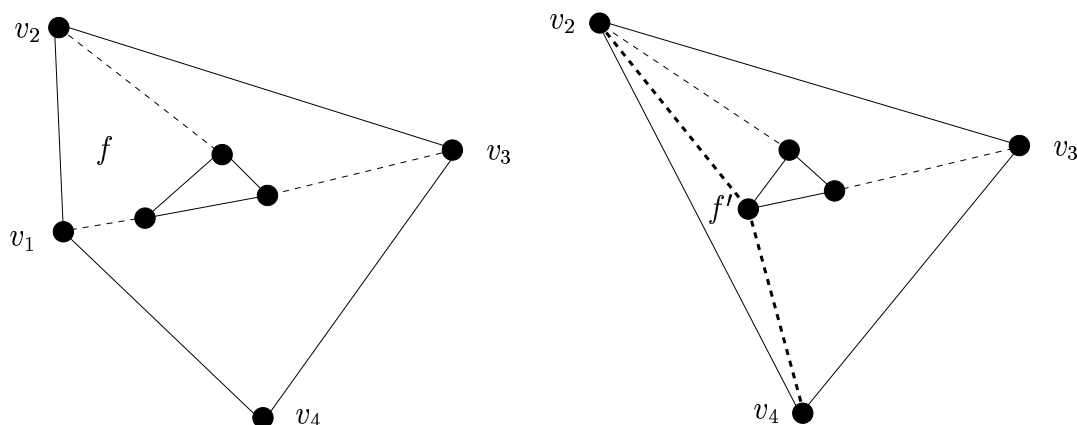


Abbildung 9.10: Vorgehen beim Herstellen der Anforderungen des 4-Sektorenmodells.

Sektorstartknoten i der Außenfläche von K_2 mit v_i verbunden. Wenn v_i nicht existiert, dann wird Sektorstartknoten i mit v_{i-1} und v_{i+1} ⁵ verbunden (in Abbildung 9.10 auf der rechten Seite sind das z.B. die beiden dicken gestrichelten Linien).

Die Attribute \preceq_x und \preceq_y der Kante von v_i zu dem jeweiligen Sektorstartknoten werden so gewählt, dass die Lagebeziehungen so wie in einem Sektor i sind. Also z.B. muss Sektorstartknoten 2 rechts unterhalb v_2 liegen.

Wir zeigen die Korrektheit der Konstruktion an zwei ausgewählten Flächenformen. Ein analoges Vorgehen ist bei allen anderen Flächen möglich.

In Fläche f wird Sektorstartknoten 2 mit Sektorstartknoten 2 verbunden (siehe Abbildung 9.10 auf der linken Seite):

Die Knoten und Kanten der Komponente K_2 , die in f liegen, liegen in f entweder ebenfalls in Sektor 2, falls in der Außenfläche von K_2 der Sektor 1 leer war, oder in Sektor 3, falls der Sektor 1 nicht leer war. Die Kante zu v_1 liegt in Sektor 3 und die übrigen Knoten und Kanten bis v_2 liegen entweder in Sektor 4 (falls v_1 nicht Sektorstartknoten 1 war) oder in Sektor 4.

⁵ bzgl. Modulo-Rechnung

Damit gibt es ein Paar von gegenüberliegenden Sektoren, die nicht leer sind und die Abfolge der Sektoren entspricht dem 4-Sektorenmodell. Das 4-Sektorenmodell ist also erfüllt.

In Fläche f' wird v_2 mit Sektorstartknoten 1 aus K_2 verbunden. Sektor 4 ist in K_1 leer und damit wird auch v_4 mit Sektorstartknoten 1 aus K_2 verbunden (siehe Abbildung 9.10 auf der rechten Seite):

Die beiden neuen Kanten liegen in f' in Sektor 2. Die übrigen Knoten und Kanten bis v_2 liegen in Sektor 4.

Damit gibt es wieder ein Paar von gegenüberliegenden Sektoren, die nicht leer sind und die Abfolge der Sektoren entspricht dem 4-Sektorenmodell.

Außerdem muss noch die Kante $e = \{v, v'\}$, die die Komponenten verbindet, berücksichtigt werden. Sei v ein Knoten in Sektor i der Komponente K_1 . Die Abbildung 9.11 illustriert die folgenden Überlegungen.

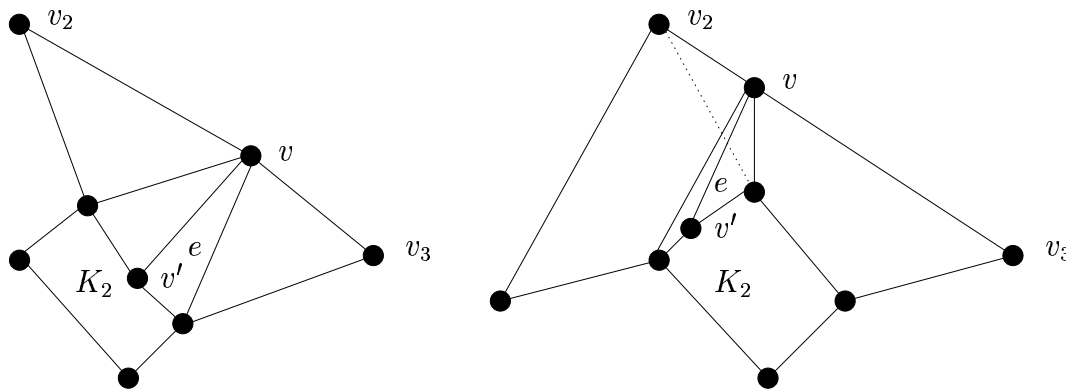


Abbildung 9.11: Verbinden der an v' angrenzenden Sektorstartknoten mit v .

Seien die Lagebeziehungen von e so gewählt, dass e „senkrecht“ in f hineinragt. Wir nehmen an, dass K_2 so angeordnet sei, dass v' in der Außenfläche von K_2 in Sektor $i - 1$, Sektor i oder Sektor $i + 1$ liegt⁶. Sei j der Sektor von v' . Dann verbinden wir den Sektorstartknoten j und den dem Sektor j folgenden Sektorstartknoten der Außenfläche von K_2 mit v . Die beiden neuen Kanten sollen dabei v in die gleiche Richtung verlassen wie e (siehe auch Abbildung 9.11). Falls $i \neq j$ gilt, dann muss noch die Kante zwischen v_i und einem Sektorstartknoten der inneren Komponente K_2 gelöscht werden, da es sonst einen Kantenschnitt gäbe.

⁶ Wenn v' in Sektor $i + 2$ läge, dann müsste e um k_2 herumgeführt werden, was so vermieden werden kann

9.3.2 Verschmelzen von zwei zweifach Zusammenhangskomponenten

Wieder wollen wir davon ausgehen, dass eine zweifach Zusammenhangskomponente in einer anderen liegt. Die beiden Komponenten seien schon durch eine Kante verbunden. Eine neue Kante, in die Einbettung planar einzufügende Kante soll die beiden Komponenten nun zu einer zweifach Zusammenhangskomponente verbinden.

Egal wie die Lagebeziehungen der neuen Kante gewählt werden, in der Regel wird das resultierende Restriktionensystem nicht mehr dem 4-Sektorenmodell entsprechen (siehe beispielsweise Abbildung 9.12). Wenn die Anforderungen des 4-Sektorenmodells wieder

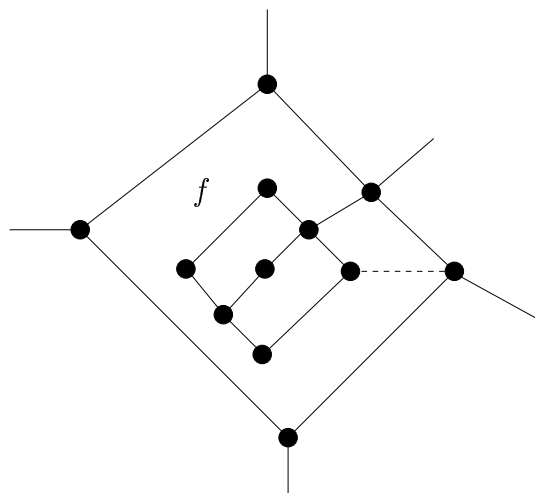


Abbildung 9.12: Zwei zweifach Zusammenhangskomponenten werden durch eine neue Kante zu einer zweifach Zusammenhangskomponente verbunden.

hergestellt werden sollen, dann müssen in der inneren zweifach Zusammenhangskomponente starke Veränderungen vorgenommen werden, denn die Außenfläche muss stark deformiert werden, damit die Fläche f und damit der Gesamtgraph mit seinem Restriktionensystem wieder dem 4-Sektorenmodell entsprechen kann.

Damit der Gesamtgraph ohne Änderungen des Restriktionensystems wieder dem 4-Sektorenmodell entsprechen würde, müsste der Anwender ähnlich zu dem Verfahren zum Zeichnen eines einfach zusammenhängenden Graphen die Sektorstartknoten der inneren Komponente geeignet mit der umschließenden Fläche verbinden.

Eine spannende Frage in diesem Zusammenhang ist es, wie viele Knicke die zusätzliche Kante maximal haben muss, damit sie planar in die gegebene Zeichnung eingebettet werden kann.

Lemma 64

Gegeben sei ein einfach zusammenhängender Graph mit einem Restriktionensystem R . R sei dergestalt, dass jede zweifach Zusammenhangskomponente dem 4-Sektorenmodell entspricht. Sei K_2 eine zweifach Zusammenhangskomponente in einer inneren Fläche f einer anderen Komponente K_1 und seien die beiden durch eine Kante e verbunden.

Eine neue Kante, die die beiden Komponenten verbindet, kann mit maximal drei Knicken gezeichnet werden. Dabei existieren Fälle, in denen die drei Knicke notwendig sind. Das 4-Sektorenmodell ist dabei für die neu entstandene zweifach Zusammenhangeskomponente nicht mehr i.A. erhalten.

BEWEIS O.B.d.A⁷ liege der Endknoten v der neuen Kante e' an der inneren Komponente in Sektor 2.

Damit der zweite Endknoten der Kante planar verbunden werden kann, muss die Kante e' entweder im Uhrzeigersinn oder im Gegenuhrzeigersinn um K_2 herumgeführt werden, da sonst e geschnitten würde. Speziell gilt, wenn der Endknoten von e und e' in K_1 identisch ist, dann kann man sowohl im Uhrzeiger- als auch im Gegenuhrzeigersinn um K_1 herumlaufen. Wir wählen in so einem Fall die Richtung, in der e' um die wenigsten Sektorstartknoten in der Außenfläche von K_2 geführt werden muss.

O.B.d.A. nehmen wir an, dass wir im Gegenuhrzeigersinn (siehe auch Abbildung 9.13) um K_1 herumlaufen müssen. Die Kante e' kann ohne Knick zu allen Knoten verbunden werden, die in f in Sektor 2 oder 1 liegen (außer, die Kanten e' und e würden sich dadurch schneiden). Diese Knoten können senkrecht über v geschoben werden, da die Knoten durch das Restriktionensystem gegenseitig in x-Richtung ungeordnet sind.

Nach einem Linksknick kann die Kante e' zusätzlich zu allen Knoten in Sektor 4 verbunden werden, denn der Knick und die Knoten sind in y-Richtung ungeordnet. Mit dem gleichen Argument kann nach zwei weiteren Linksknicken die Kante zu allen Knoten in Sektor 3 und dann zu den übrigen Knoten in Sektor 2 verbunden werden.

Damit sind in Summe drei Kantenknicke hinreichend, um jeden Knoten zu erreichen.

Die Abbildung 9.13 ist gleichzeitig auch ein Beispiel, in dem drei Knicke notwendig sind. ■

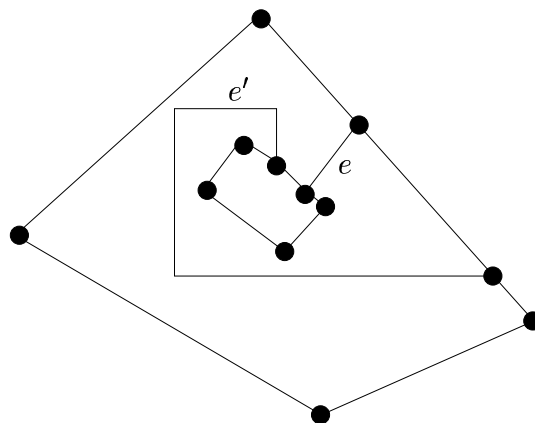


Abbildung 9.13: Eine Kante, die den zweifach Zusammenhang herstellt, kann mit drei Knicken gezeichnet werden.

⁷ Alle anderen Fälle lassen sich durch Spiegelungen daraus ableiten.

Das Ergebnis aus diesem Abschnitt ist in Summe eher negativ. Einfach zusammenhängende Graphen können in den Ansatz integriert werden. Allerdings führt das Verschmelzen von zwei einfach Zusammenhangskomponenten zu einer zweifach Zusammenhangskomponente dazu, dass entweder das 4-Sektorenmodell nicht mehr beachtet wird, oder der Graph muss umfangreich geometrisch angepasst werden, damit das 4-Sektorenmodell wieder erfüllt ist.

9.4 Nicht planares Einfügen von Kanten

In diesem Abschnitt behandeln wir das nicht planare Einfügen von Kanten. Dabei wollen wir uns hier nicht mit der Frage beschäftigen, wie z. B. die Anzahl der Kantenschnitte minimiert werden könnte. Wir wollen nur zeigen, dass auch nicht-planare Graphen mit unserem Ansatz vereinbar sind, denn dies ist eine Anforderung aus realen Anwendungen, wie z.B. dem Zeichnen von Objektmodellen oder UML-Diagrammen, die in der Regel nicht planare Graphen darstellen.

Wir setzen voraus, dass eine Kante keinen Knoten schneiden darf.

In [FK97] zeigen Fößmeier und Kaufmann, wie der Kandinsky-Algorithmus um nicht planare Graphen erweitert werden kann. Sie zeigen dabei auch, dass es nicht sinnvoll ist, auf jeden Kantenschnitt einfach nur einen Knoten zu legen und dann den normalen Kandinsky-Algorithmus anzuwenden.

Dieses Vorgehen würde auf Grund der „bend-or-end“ Eigenschaft des Kandinsky-Modells zu unnötigen Kantenknicken führen (Abbildung 9.14). In Abbildung 9.14 auf der rechten Seite ist eine Zeichnung dargestellt, die das Ergebnis eines intelligenteren Algorithmus ist. Da unser Ansatz ebenfalls auf dem Kandinsky-Modell basiert, kann das

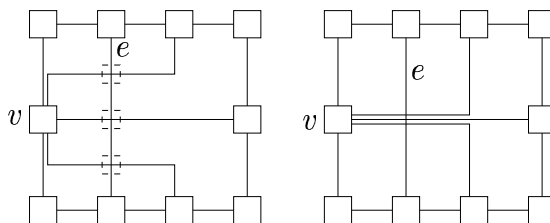


Abbildung 9.14: Eine einfache Integration von nicht planaren Graphen in den Kandinsky-Algorithmus führt zu unnötigen Kantenknicken.

Beispiel auf unser Verfahren übertragen werden.

Außerdem erhält bei einer Zeichnung ohne Knickoptimierung jede Kante mindestens einen Knick, und damit auch jedes Teilstück einer Kante, die durch mehrere Flächen verläuft. Auch die geschnittenen Kanten des eigentlich planaren Graphen erhalten auf diese Art zusätzliche Kantenknicke. (siehe Abbildung 9.15 für ein Beispiel. Auch hier ist wieder auf der linken Seite die direkte Integration mit Dummy-Knoten und rechts ein möglicher Kantenverlauf dargestellt.)

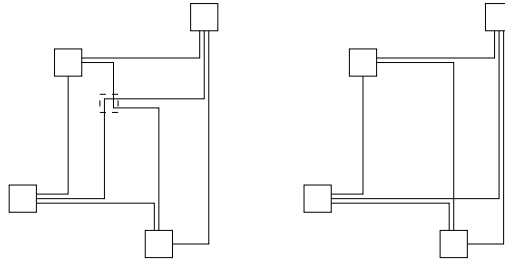


Abbildung 9.15: Eine einfache Integration von nicht planaren Graphen in unser Verfahren führt zu unnötigen Kantenknicken.

Das Einfügen von Dummyknoten führt also zu vielen unnötigen Kantenknicken. Deshalb stellen wir hier einen anderen Ansatz vor:

Gegeben sei ein Graph $G = (V, E)$ mit einem legalen Restriktionensystem R und der Optimierungsgraph G_{opt} . Außerdem sei eine Menge E' von Kanten gegeben, bei denen jede Kante beim Einfügen in die gegebene planare Einbettung von G Kantenschnitte verursacht.

Das Ziel ist es, die Kanten aus E' so einzufügen, dass sie genau einen Knick haben und beim linken Endknoten zunächst mit einem waagerechten Kantensegment beginnen, gefolgt von einem senkrechten. Die Kanten sollen also genau den gleichen Verlauf haben wie die Kanten aus E , wenn sie mit einem Knick gezeichnet werden.

1. Es muss sichergestellt werden, dass auf den Seiten eines Knotens v , an denen eine Kante aus $e' \in E'$ beginnt, kein Knoten liegen darf, der von e' geschnitten würde. Das ist dann der Fall, wenn eine Kante ohne Knick auf dieser Seite beginnt und der Endknoten der Kante ohne Knick näher an v liegen muss, als der zweite Endknoten der Kante e' .

Deshalb muss in so einem Fall der Knoten aus G_{opt} gelöscht werden, der das Zeichnen der Kante ohne Knick symbolisiert.

2. Nach dem normalen Verfahren (wenn nötig auch unter Berücksichtigung aller Mindestabstände für variable Knotengrößen und Kantenbeschriftungen) wird die Knickoptimierung und eine Zuordnung der Knoten von G an Zeilen und Spalten berechnet.
3. In diesem Schritt soll sichergestellt werden, dass Kantenbeschriftungen nicht durch die Kanten aus E' geschnitten werden. Dieser Schritt ist deshalb nur im Fall von Kantenbeschriftungen nötig.

Es wird berechnet, welche Kanten aus E durch die Kanten aus E' geschnitten werden. Schnittpunkte werden temporär durch Dummyknoten ersetzt. Nach dem normalen Verfahren unter Berücksichtigung aller Mindestabstände für variable Knotengrößen und Kantenbeschriftungen wird eine Zuordnung der Knoten von G an Zeilen und Spalten berechnet.

Die Dummyknoten werden wieder gelöscht. Kanten aus E' können in den Zeilen und Spalten platziert werden, in denen die Dummyknoten lagen.

4. Es muss sichergestellt werden, dass kein Knoten $v \in V$ durch eine Kante aus E' geschnitten wird (Abbildung 9.16 zeigt ein Beispiel). Der Unterschied zu Schritt 1 ist, dass diese Knoten nicht notwendigerweise zu den Endknoten der Kanten aus E' adjazent sein müssen.

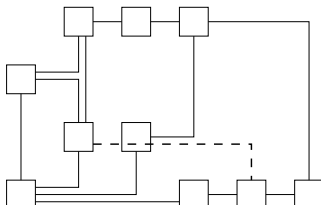


Abbildung 9.16: Knoten können in einer Zeile oder Spalte liegen, in der sie von einer Kante aus E' geschnitten werden.

Wir durchlaufen alle Kanten $e' = \{v_1, v_2\} \in E'$. Sei v_1 der linke Endknoten. Wenn ein Knoten aus V durch den waagerechten Teil von e' geschnitten wird, dann erzeugen wir eine neue Zeile unter der Zeile von v_1 . Alle Knoten, die rechts von v_1 in der gleichen Zeile liegen, werden in die neue Zeile gelegt. Wir legen die ganze Menge an Knoten in die neue Zeile, weil der geschnittene Knoten der linke Endknoten einer waagerechten Kante ohne Knick sein könnte. Auf diese Art können wir einfach sicherstellen, dass alle waagerechten Kanten auch waagerecht bleiben. Wenn wir sicherstellen wollen, dass bei einer unter Umständen anschließend ausgeführten Kompaktierung die Knoten u , die gerade in die neue Zeile geschoben wurden, auch unterhalb von e' verbleiben, dann muss eine zusätzliche Lageabhängigkeit zwischen u und v_1 eingefügt werden, die sicherstellt, dass u unter v_1 liegt.

Entsprechend verfahren wir mit allen Knoten, die von dem senkrechten Teil einer Kante aus E' geschnitten werden.

5. Die Zeichnung des Graphen G gemeinsam mit den Kanten aus E' wird nach dem üblichen Schema berechnet.

Mit etwas intelligenteren Verfahren können zum Teil auch die Knicke in den Kanten aus $e' \in E'$ eingespart werden. In Schritt 1 haben wir willkürlich einen der Endknoten der Kante e' als linken Endknoten festgelegt, wenn die beiden Knoten in x-Richtung ungeordnet sind. In diesem Fall könnten die beiden Endknoten aber auch in einer gemeinsamen Spalte liegen und es wäre möglich, die Kante e' senkrecht ohne Knick zu zeichnen. Entsprechendes gilt, wenn die beiden Knoten in y-Richtung ungeordnet sind.

Das vorgestellte Verfahren bietet eine einfache Möglichkeit zur Erweiterung unseres Algorithmus auf nicht planare Graphen. Ein kleiner Eingriff in die Knickoptimierung stellt sicher, dass kein Knoten durch eine Kante geschnitten wird. Dafür kann man in

Summe mehr Kantenknicke erhalten. Kanten aus E' erhalten immer nur einen Knick, was insbesondere deshalb ein Vorteil ist, weil die Endknoten zwischen Knoten in verschiedenen Flächen in der Regel weiter auseinanderliegen und die Kanten wegen der Kantenschnitte nur schwierig zu verfolgen sind.

Es wird bei dem Verfahren keine Rücksicht auf die Summe der Kantenschnitte genommen. Dies könnte verbessert werden, wenn ein Verlauf berechnet wird, der möglichst wenige Flächen schneidet. Allerdings müsste dann ein komplexerer Kantenverlauf berechnet werden und die Kante $e' \in E'$ könnte mehr als einen Knick erhalten. Als Beispiel dazu nehmen wir an, dass die Kante eine Fläche f in Sektor 3 betritt und auch wieder verlässt. In diesem Fall müsste e' in f einen Knick haben (siehe auch Abbildung 9.17).

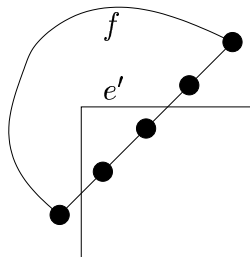


Abbildung 9.17: Kante e' muss in f einen Knick haben, wenn die Fläche in Sektor 3 betreten und wieder verlassen wird.

In [BM97] zeigen Biedl und Kaufmann einen Algorithmus zum interaktiven Zeichnen von Graphen im Kandinsky-Modell. Sie erhalten in jeder Kante maximal einen Knick, in Summe also maximal m . Wir erhalten in Summe maximal $m + |F|$ Kantenknicke. Die zusätzlichen $|F|$ Kantenknicke ergeben sich bei uns aus dem Algorithmus zum planaren Zeichnen des Graphen.

Damit haben wir ein Verfahren vorgestellt, mit dem auch nicht planare Graphen in unseren Algorithmus integriert werden können.

Im 4-Sektorenmodell, gemeinsam mit unseren Anforderungen, dass die Einbettung bei den Einfügeoperationen nicht geändert werden darf, kann es auch bei eigentlich planaren Graphen passieren, dass wir Schnitte zwischen Kanten erhalten. Das ist dann der Fall, wenn es eine Einbettung gibt, so dass der Graph planar gezeichnet werden kann, aber eine andere Einbettung festgelegt ist. In diesen Fällen können die neuen Kanten genauso eingefügt werden, wie eine Kante, mit der keine planare Einbettung existiert.

Damit sind alle im Rahmen dieser Arbeit betrachteten Erweiterungen beschrieben. Wir wollen nun noch in einem letzten Kapitel alle Ergebnisse zusammenfassen und weitere Fragestellungen beleuchten.

10 Zusammenfassung und Ausblick

Wir wollen zunächst die Ergebnisse dieser Arbeit nochmal zusammenfassen und dann die noch offenen Fragen erläutern.

Die zugrunde liegende Idee war, dass zunächst ein Verfahren geschaffen wird, das mit einfachen Algorithmen, Datenstrukturen und Invarianten arbeitet. Dafür haben wir auch eine Einschränkung der betrachteten Graphklasse in Kauf genommen, in dem wir uns zunächst nur mit zweifach zusammenhängenden planaren Graphen beschäftigt haben. Wir haben dafür ein Verfahren entwickelt, das einfach und modular um zusätzliche Komponenten erweitert werden kann.

Die zum Einfügen verwendeten Operationen benötigen dabei $O(\log n)$ Laufzeit, der Speicher ist durch $O(n)$ beschränkt. Wir haben das durch eine geschickte Verwendung der Eigenschaften des 4-Sektorenmodells mit Hilfe von balancierten Bäumen erreicht. Das Zeichnen des Graphen erfolgt in Linearzeit.

Die einfachen Invarianten unseres Verfahrens sind die Einbettung und die Lagebeziehungen zwischen adjazenten Knoten, erweitert um festgelegte Kantenverläufe. Wir erhalten dadurch mehr Kantenknicke, als unbedingt nötig wären. Allerdings sind sich zwei aufeinander folgende Zeichnungen wesentlich ähnlicher, wenn die Kantenverläufe nicht beliebig variiert werden dürfen.

Wir haben das Optimierungsproblem, das sich ergibt, wenn die Kantenverläufe nicht einschränkt werden und bei einer $<$ -Relation möglichst viele Kanten mit nur einem Knick gezeichnet werden sollen, im Rahmen der Arbeit nicht weiter untersucht. Statt dessen haben wir uns mit der Frage beschäftigt, wie in unserem Modell die Summe der Kantenknicke minimiert werden kann, falls wir \leq als Ordnungsrelation verwenden. Eine Kante darf nur dann den festgelegten Verlauf ändern, wenn dadurch bei der Kante ein Knick eingespart werden kann. Wir haben eine Lösung unter Verwendung von „maximum independent set“ für bipartite Graphen gefunden und erhalten damit eine $O(n^2 \log n)$ Laufzeit für die Optimierung und erhalten eine bzgl. unserer Einschränkungen knickoptimale Zeichnung. Die allgemeine Frage, ob bei einer Einschränkung der erlaubten Zeichnungen durch die Angabe von Lagebeziehungen in Polynomzeit möglich ist, ist offen.

Anschließend haben wir die Knickoptimierung dynamisiert. Das heißt, es können die in der letzten Zeichnung verwendeten Knickoptimierungen größtenteils beibehalten werden, was die Ähnlichkeit zwischen den Zeichnungen erhöht.

Unserem Restriktionensystem, gemeinsam mit der Einschränkung der erlaubten Kantenverläufe und unter Umständen den Mengen E^s , E^w und E^1 , entspricht eindeutig eine orthogonale Repräsentation. Aufgrund der Eigenschaften unseres Zeichenmodells kann

dafür mit wenig Aufwand eine Zeichnung generiert werden. Ein geeigneter Algorithmus wurde angegeben. Wir haben zusätzlich gezeigt, dass das Zeichnen mit minimaler Fläche auch für unser eingeschränktes Modell NP-schwer ist.

Unser Zeichenalgorithmus kann auch dann verwendet werden, wenn die Größe eines Knotens Teil der Benutzereingabe ist. Wir haben uns außerdem mit der Frage beschäftigt, wie die betrachtete Graphklasse erweitert werden kann. Dafür haben wir zunächst einfach zusammenhängende und dann nicht planare Graphen in das Verfahren integriert. Beide Erweiterungen führen wir wieder auf zweifach zusammenhängende planare Graphen mit einem Restriktionensystem im 4-Sektorenmodell zurück. Im Falle von nicht planaren Graphen müssen die Kanten, die die Planarität zerstören, nachträglich eingezeichnet werden. Allerdings ist ihr Verlauf schon in der restlichen Zeichnung berücksichtigt, also der nötige freie Raum bereits geschaffen. Beim einfachen Zusammenhang ist es aber bei einer Verschmelzung von einem vorher einfach zusammenhängenden Graphen zu einem zweifach zusammenhängenden nicht möglich, das Restriktionensystem zu erhalten, wenn die Invarianten des 4-Sektorenmodells wieder hergestellt werden sollen.

Schließlich haben wir uns noch mit einer Anforderung aus realen Anwendungen beschäftigt, in dem wir Kantenbeschriftungen in unser Verfahren integriert haben.

Alle Erweiterungen, auch die Knickoptimierung, benutzen begrenzte Informationen, wie z. B. die orthogonale Repräsentation, und führen lokale Veränderungen durch oder ergänzen bestimmte Attribute wie minimale Kantenlänge. Attribute sind jeweils so aufgebaut, dass sie leicht in die Erzeugung der orthogonalen Repräsentation oder der Zeichnung integriert werden können. Die notwendigen Veränderungen an Datenstrukturen wie z. B. der Einbettung wurden so durchgeführt, dass sie wieder die Invarianten des 4-Sektorenmodells erfüllen. Alle anderen Teile bleiben davon unberührt und können in beliebiger Art und Weise kombiniert werden.

Auf Grund dieses strengen Aufbaus in unabhängige Komponenten sollten auch zusätzliche Erweiterungen relativ leicht integrierbar sein. Wir wollen hier in einem kleinen Ausblick die aus unserer Sicht wichtigsten Ergänzungen noch kurz ansprechen. Außerdem sprechen wir noch interessante theoretische Fragen zu den schon vorgestellten Komponenten an.

1. Jeder Mensch macht Fehler. Deshalb wäre es sinnvoll, eine Funktion zum Löschen von Knoten und Kanten zu integrieren.

Nach dem Löschen müssen die Invarianten des 4-Sektorenmodells wieder hergestellt werden. Ähnlich wie beim einfachen Zusammenhang muss dabei im Allgemeinen das Restriktionensystem verändert werden.

Dabei stellt sich die Frage, ob die notwendigen Änderungen am Restriktionensystem lokal eingegrenzt werden können und wie komplex die entsprechende Operation ist.

2. Wir haben gezeigt, dass das Zeichnen mit minimaler Fläche NP-schwer ist. Die Heuristik, die wir dann zum Zeichnen verwenden, geht je nach Restriktionensystem

sehr verschwenderisch mit der Fläche um. Hier würde sich die Suche nach einer besseren Heuristik lohnen.

3. Bei Einfügen von Kanten, die die Planarität zerstören, haben wir gezeigt, dass diese Kanten jeweils mit nur einem Knick gezeichnet werden können. Kann man alle Verläufe durch die anderen Flächen, die eine Zeichnung mit einem Knick erlauben, irgendwie charakterisieren? Wenn ja, findet man unter dieser Menge von Verläufen in Polynomzeit einen Verlauf, der die minimale Anzahl an Kantenschnitten hat?
4. Oft betrachtet man nicht nur flache Graphen, sondern auch hierarchische. Kann man die Ideen für Graph Grammatiken aus [Bra94] an dieser Stelle verwenden? Dazu muss man für „versteckte“, also im Augenblick nicht dargestellte, Komponenten geeignete Lagebeziehungen finden.
5. Die Einbettung haben wir nie geändert. Wir haben sie auch dann so gehalten, wenn wir dadurch unnötige Kantenschnitte erhalten haben. Aus praktischer Sicht erscheint uns das sehr sinnvoll. Trotzdem wäre es schön, eine Änderung der Einbettung in das Verfahren zu integrieren, die dann nötig wird, wenn zwei Knoten in verschiedenen Flächen durch eine Kante verbunden werden. Mit Hilfe von *SPQR*-Bäumen sollte das möglich sein. Hier stellt sich dann die Frage, ob man eine minimale Änderung des Restriktionensystems charakterisieren und finden kann.

Literaturverzeichnis

- [AHU83] AHO, A. V., J. E. HOPCROFT und J. D. ULLMAN: *Data Structures and Algorithms*. Addison-Wesley, 1983.
- [Bac02] BACHL, W.: *Semi-dynamic Orthogonal Drawings of Planar Graphs*. In: *Graph Drawing GD '02*, LNCS 2528, S. 354–361, 2002.
- [BBD⁺99] BRIDGEMAN, S. S., G. DI BATTISTA, W. DIDIMO, G. LIOTTA, R. TAMASSIA und L. VISMARA: *Turn-Regularity and Planar Orthogonal Drawings*. In: *Graph Drawing GD '99*, LNCS 1731, S. 8–26, 1999.
- [BETT98] BATTISTA, G. DI, P. EADES, R. TAMASSIA und I. G. TOLLIS: *Graph Drawing*. Prentice Hall, 1998.
- [BFG⁺97] BRIDGEMAN, S.S., J. FANTO, A. GARG, R. TAMASSIA und L. VISMARA: *InteractiveGiotto: An Algorithm for Interactive Orthogonal Graph Drawing*. In: *Graph Drawing GD '97*, LNCS 1353, S. 303–308, 1997.
- [BGW00] BRANDES, U., M. GÜDEMANN und D. WAGNER: *Fully Dynamic Orthogonal Graph Layout for Interactive Systems*. Technischer Bericht 111, Universität Konstanz, 2000.
- [Bie95] BIEDL, T. C.: *New Lower Bounds for Orthogonal Graph Drawing*. In: *Graph Drawing GD '95*, LNCS 1027, S. 28–39, 1995.
- [Bie97] BIEDL, T. C.: *Orthogonal Graph Visualization: The Three-Phase Method with Applications*. Doktorarbeit, Rutgers, The State University of New Jersey, 1997.
- [Bie98] BIEDL, T. C.: *Three Approaches to 3D-Orthogonal Box-Drawings*. In: *Graph Drawing GD '98*, LNCS 1547, S. 30–43, 1998.
- [BM97] BIEDL, T. und M. KAUFMANN: *Area-Efficient Static and Incremental Graph Drawings*. In: *Proc. 5th European Symposium on Algorithms (ESA '97)*, LNCS 1284, S. 37–52, 1997.
- [BMT97] BIEDL, T. C., B. MADDEN und I. G. TOLLIS: *The Three-Phase Method: A Unified Approach to Orthogonal Graph Drawing*. In: *Graph Drawing GD '97*, LNCS 1353, S. 391–402, 1997.

- [BPV98] BATTISTA, G. DI, M. PATRIGNANI und F. VARGIU: *A Split&Push Approach to 3D Orthogonal Drawing*. In: *Graph Drawing GD '98*, LNCS 1547, S. 87–101, 1998.
- [Bra94] BRANDENBURG, F. J.: *Designing graph drawings by layout graph grammars*. In: *Graph Drawing GD '94*, LNCS 894, S. 416–427, 1994.
- [Bra99] BRANDES, U.: *Layout of Graph Visualizations*. Doktorarbeit, Universität Konstanz, Fakultät für Mathematik und Informatik, 1999.
- [BSWW97] BIEDL, T. C., T. SHERMER, S. WHITESIDES und S. WISMATH: *Orthogonal 3-D Graph Drawing*. In: *Graph Drawing GD '97*, LNCS 1353, S. 76–86, 1997.
- [BT96] BATTISTA, G. DI und R. TAMASSIA: *On-Line Planarity Testing*. *SIAM Journal on Computing*, 25(5):956–997, 1996.
- [BT98] BRIDGEMAN, S. und R. TAMASSIA: *Difference Metrics for Interactive Orthogonal Graph Drawing Algorithms*. In: *Graph Drawing GD '98*, LNCS 1547, S. 57–71, 1998.
- [BW97] BRANDES, U. und D. WAGNER: *A Bayesian Paradigm for Dynamic Graph Layout*. In: *Graph Drawing GD '97*, LNCS 1353, S. 236–247, 1997.
- [BW98] BRANDES, U. und D. WAGNER: *Dynamic Grid Embedding with Few Bends and Changes*. In: *9th Annual International Symposium on Algorithms and Computation (ISAAC'98)*, S. 89–98, 1998.
- [CBT⁺92] COHEN, R. F., G. DI BATTISTA, R. TAMASSIA, I. G. TOLLIS und P. BERTOLAZZI: *A Framework for Dynamic Graph Drawing*. In: *Proceedings of the 8th Annual Symposium on Computational Geometry (SCG '92)*, S. 261–270, 1992.
- [CBTT95] COHEN, R. F., G. DI BATTISTA, R. TAMASSIA und I. G. TOLLIS: *Dynamic Graph Drawings: Trees, Series-Parallel Digraphs, and Planar ST-Digraphs*. *SIAM Journal on Computing*, 24(5):970–1001, 1995.
- [DL97] DIDIMO, W. und A. LEONFORTE: *GRID: An Interactive Tool for Computing Orthogonal Drawings With the Minimum Number of Bends*. In: *Graph Drawing GD '97*, LNCS 1353, S. 309–315, 1997.
- [EF97] EADES, P. und Q. FENG: *Drawing Clustered Graphs on an Orthogonal Grid*. In: *Graph Drawing GD '97*, LNCS 1353, S. 146–157, 1997.
- [EFK00] EIGLSPERGER, M., U. FÖSSMEIER und M. KAUFMANN: *Orthogonal Graph Drawing with Constraints*. In: *Soda 2000*, 2000.
- [Eig99] EIGLSPERGER, M.: *Orthogonales Graphenzeichnen mit Constraints*. Diplomarbeit, Universität Tübingen, 1999.

-
- [ELMS91] EADES, P., W. LAI, K. MISUE und K. SUGIYAMA: *Preserving the mental map of a diagram*. In: *COMPUGRAPHICS '91*, S. 34–43, 1991.
- [ESW96] EADES, P., A. SYMVONIS und S. WHITESIDES: *Two Algorithms for Three Dimensional Orthogonal Graph Drawing*. In: *Graph Drawing GD '96*, LNCS 1190, S. 139–154, 1996.
- [FHK98] FÖSSMEIER, U., C. HESS und M. KAUFMANN: *On Improving Orthogonal Drawings: The 4M-Algorithm*. In: *Graph Drawing GD '98*, LNCS 1547, S. 125–137, 1998.
- [FK94] FÖSSMEIER, U. und M. KAUFMANN: *On Bend-Minimum Orthogonal Upward Drawings of Directed Planar Graphs*. In: *Graph Drawing GD '94*, LNCS 894, S. 52–63, 1994.
- [FK95] FÖSSMEIER, U. und M. KAUFMANN: *Drawing High Degree Graphs with Low Bend Numbers*. In: *Graph Drawing GD '95*, LNCS 1027, S. 254–266, 1995.
- [FK97] FÖSSMEIER, U. und M. KAUFMANN: *Algorithms and Area Bounds for Non-planar Orthogonal Drawings*. In: *Graph Drawing GD '97*, LNCS 1353, S. 134–145, 1997.
- [FKK96] FÖSSMEIER, U., G. KANT und M. KAUFMANN: *2-Visibility Drawings of Planar Graphs*. In: *Graph Drawing GD '96*, LNCS 1190, S. 155–168, 1996.
- [Fös97] FÖSSMEIER, U.: *Orthogonale Visualisierungstechniken für Graphen*. Doktorarbeit, Universität Tübingen, 1997.
- [Föß97] FÖSSMEIER, U.: *Interactive Orthogonal Graph Drawing: Algorithms and Bounds*. In: *Graph Drawing GD '97*, LNCS 1353, S. 111–123, 1997.
- [GJ83] GAREY, M. R. und D. S. JOHNSON: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1983.
- [GM98] GUTWENGER, C. und P. MUTZEL: *Planar Polyline Drawings with Good Angular Resolution*. In: *Graph Drawing GD '98*, LNCS 1547, S. 167–182, 1998.
- [GT94] GARG, A. und R. TAMASSIA: *On the Computational Complexity of Upward and Rectilinear Planarity Testing*. In: *Graph Drawing GD '94*, LNCS 894, S. 286–297, 1994.
- [GT96] GARG, A. und R. TAMASSIA: *A new Minimum Cost Flow Algorithm with Applications to Graph Drawing*. In: *Graph Drawing GD '96*, LNCS 1190, S. 201–216, 1996.
- [Har74] HARARY, F.: *Graphentheorie*. Oldenbourg Verlag, 1974.
-

- [Imh75] IMHOF, E.: *Positioning Names on Maps*. The American Cartographer, S. 128–144, 1975.
- [Kan96] KANT, G.: *Drawing Planar Graphs using the Canonical Ordering*. Algorithmica, 16(1):4–32, 1996.
- [Kan97] KANT, G.: *A more compact visibility representation*. International Journal of Computational Geometry & Applications, 7(3):197–210, 1997.
- [KM92] KASHIWABARA, T. und S. MASUDA: *Generation of Maximum Independent Sets of a Bipartite Graph and Maximum Cliques of a Circular-Arc Graph*. Journal of Algorithms, 13, 1992.
- [KM98] KLAU, G. W. und P. MUTZEL: *Quasi-Orthogonal Drawing of Planar Graphs*. Technischer Bericht MPI-I-98-1-013, Max-Planck-Institut für Informatik, 1998.
- [KM99a] KLAU, G. W. und P. MUTZEL: *Combining Graph Labeling and Compaction*. In: *Graph Drawing GD '99*, LNCS 1731, S. 27–37, 1999.
- [KM99b] KLAU, G. W. und P. MUTZEL: *Optimal Compaction of Orthogonal Grid Drawings*. In: *International IPCO Conference (Integer Programming and Combinatorial Optimization)*, S. 304–319, 1999.
- [KT96] KAKOULIS, K. G. und I. G. TOLLIS: *On the Edge Label Placement Problem*. In: *Graph Drawing GD '96*, LNCS 1190, S. 241–256, 1996.
- [MELS95] MISUE, K., P. EADES, W. LAI und K. SUGIYAMA: *Layout Adjustment and the Mental Map*. Journal Visual Languages and Computing, 6:183–210, 1995.
- [Nor95] NORTH, S.: *Incremental Layout in DynDAG*. In: *Graph Drawing GD '95*, LNCS 1027, S. 409–418, 1995.
- [Pat99] PATRIGNANI, M.: *On the Complexity of Orthogonal Compaction*. In: *6th International Workshop Algorithms Data Struct. (Proc. WADS '99)*, LNCS 1663, S. 56–61, 1999.
- [PCJ95] PURCHASE, H. C., R. F. COHEN und M. JAMES: *Validating graph drawing aesthetics*. In: *Graph Drawing GD '95*, LNCS 1027, S. 435–446, 1995.
- [PST96] PAPAKOSTAS, A., J. M. SIX und I. G. TOLLIS: *Experimental and Theoretical Results in Interactive Orthogonal Graph Drawing*. In: *Graph Drawing GD '96*, LNCS 1190, S. 371–386, 1996.
- [PT94] PAPAKOSTAS, A. und I. G. TOLLIS: *Improved Algorithms and Bounds for Orthogonal Drawings*. In: *Graph Drawing GD '94*, LNCS 894, S. 40–51, 1994.

-
- [PT95] PAPAKOSTAS, A. und I. G. TOLLIS: *Issues in Interactive Orthogonal Graph Drawing*. In: *Graph Drawing GD '95*, LNCS 1027, S. 419–430, 1995.
- [PT96] PAPAKOSTAS, A. und I. G. TOLLIS: *A Pairing Technique for Area-Efficient Orthogonal Drawings*. In: *Graph Drawing GD '96*, LNCS 1190, S. 355–370, 1996.
- [RT86] ROSENSTIEHL, R. und R.E. TARJAN: *Rectilinear planar layouts and bipolar orientations of planar graphs*. *Journal Discrete and Computational Geometry*, S. 321–341, 1986.
- [San95] SANDER, G.: *A Fast Heuristic for Hierarchical Manhattan Layout*. In: *Graph Drawing GD '95*, LNCS 1027, S. 447–458, 1995.
- [Sch01] SCHREIBER, F.: *Visualisierung biochemischer Reaktionswege*. Doktorarbeit, Universität Passau, 2001.
- [STM81] SUGIYAMA, K., S. TAGAWA und M.TODA: *Methods for visual understanding of hierarchical systems*. In: *IEEE Transactions on Systems, Man and Cybernetics*, 1981.
- [Sto83] STOCKMEYER, L.: *Optimal Orientations of Cells in Slicing Floorplan Design*. *Information and Control*, 57(1):91–101, 1983.
- [Tam87] TAMASSIA, R.: *On embedding a graph in the grid with the minimum number of bends*. *SIAM Journal on Computing*, 16(3):421–444, 1987.
- [TBC97] TUNKELANG, D., R. J. BYRD und J. W. COOPER: *Lexical Navigation: Using Incremental Graph Drawing for Query Refinement*. In: *Graph Drawing GD '97*, LNCS 1353, S. 316–321, 1997.
- [TdBB88] TAMASSIA, R., G. DI BATTISTA und C. BATINI: *Automatic Graph Drawing and Readability of Diagrams*. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):61–79, 1988.
- [TT86] TAMASSIA, R. und I. G. TOLLIS: *A unified approach to visibility representations of planar graphs*. *Discrete and Computational Geometry 1*, S. 321–341, 1986.
- [Woo98] WOOD, D. R.: *An Algorithm for Three-Dimensional Orthogonal Graph Drawing*. In: *Graph Drawing GD '98*, LNCS 1547, S. 332–346, 1998.
- [Yoe72] YOELI, P.: *The Logic of Automated Map Lettering*. *The Cartographic Journal*, S. 99–108, 1972.